

El mundo de la COMPUTACION

Curso teórico-práctico

4

OCEANO

El mundo de la COMPUTACION

El mundo de la COMPUTACION

Curso teórico-práctico

4

OCEANO

Es una obra del

GRUPO EDITORIAL OCEANO

Presidente
José Lluis Monreal

Director General
José M.ª Martí

Director General de Publicaciones
Carlos Gispert

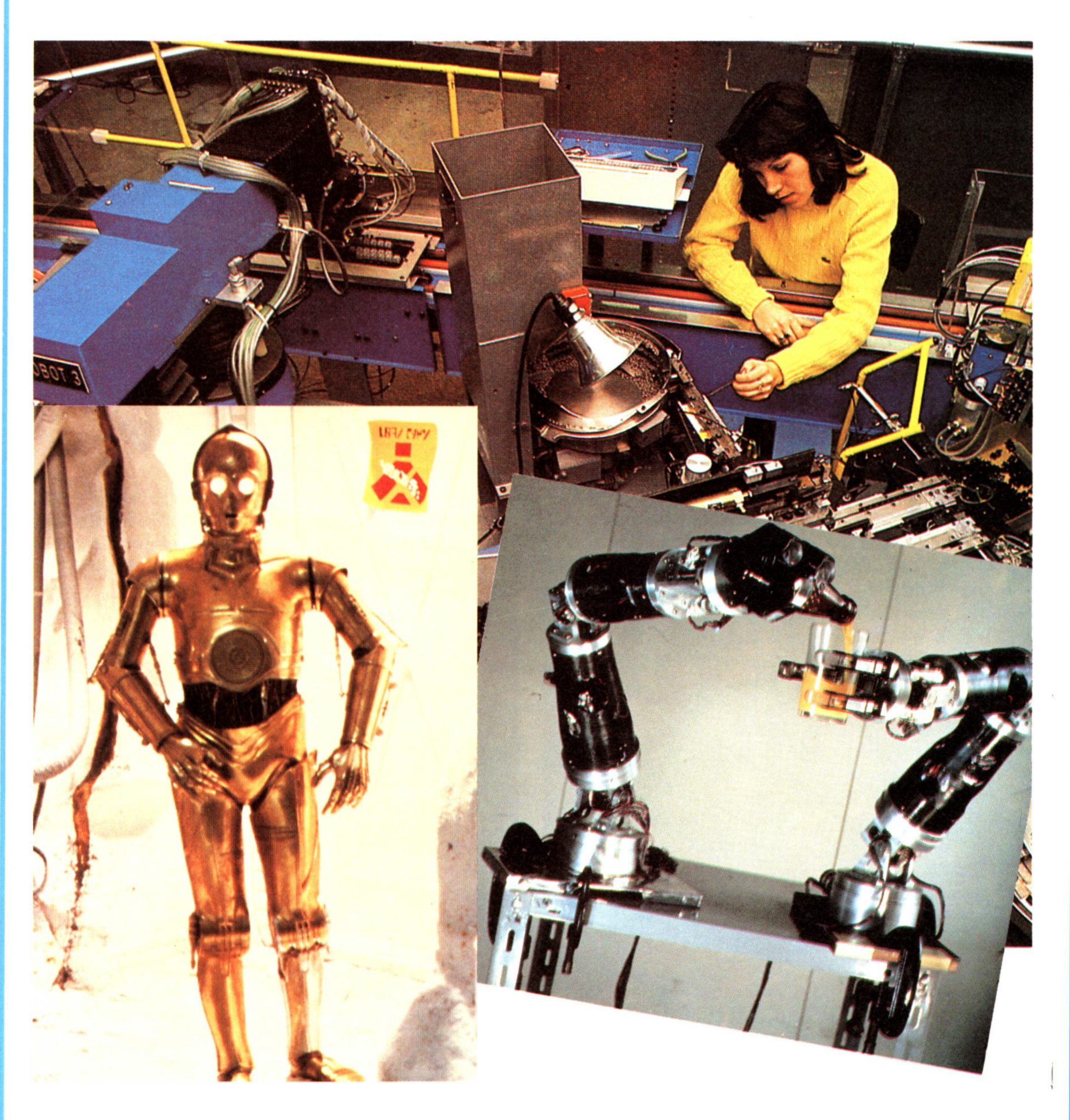
Paseo de Gracia, 24
Teléfonos: 317 45 08*
Télex: 51735 exit e - Fax: 317 97 01
08007 Barcelona (España)

Impreso en España - Printed in Spain

ISBN: 84-7764-189-7 (Obra completa)
ISBN: 84-7764-193-5 (Volumen 4)

Depósito legal: CO-878-92 (Oc)
Impreso y encuadernado por:
GRAFICROMO, S.A. Córdoba (España)

volumen 4



La computadora: una máquina para enseñar
y aprender. Programas educativos
La inteligencia artificial y la quinta generación
Instrucciones más comunes del lenguaje BASIC
Glosario
Vocabulario inglés-español

La computadora: una máquina para enseñar y aprender

Lo que hasta hace poco se tenía por el saber central de un área profesional, de un ámbito técnico o cultural de la vida, con el tiempo ha ido quedando anticuado; ya no es útil. Del poder de los ancianos se ha ido pasando, paulatinamente, al poder de una juventud madura que ha salido de un centro de formación y ha conjugado la preparación con la experiencia. Los ciclos de validez de los conocimientos se acortan cada vez más. Pasados quince años, todo el saber de un profesional queda obsoleto; si no se recicla, sus conocimientos se hallan desfasados respecto a los avances producidos.

Ya no existe una preparación definitiva, sino meramente transitoria. El papel del sistema de educación ha cambiado y, en todo ello, la escuela desempeña una función básica. La escuela responde a la necesidad de instruir y preparar a los jóvenes para la vida profesional y laboral de adultos. Es la institución de continuidad social. Y hasta hace muy poco se creía firmemente que aquello que se impartía en los recintos académicos tenía un valor vital para el alumnado. Se hacía el énfasis en los contenidos, en las ideas que había que conocer y recordar. Con ese magma mental, repleto de citas literarias, fórmulas matemáticas y fisicoquímicas, mapas geopolíticos, personajes históricos, etc., el alumno podría recorrer el mundo futuro perfectamente pertrechado. La escuela era el almacén que impartía vituallas conceptuales concretas.

La función de la escuela en la actualidad

Pero ya no es así. La función de la escuela no consiste en proveer de formas fijas de conocimiento, sino de fórmulas o métodos para resolver situaciones diversas y cambiantes. Ya no importan tanto los contenidos como los instrumentos y los métodos para llegar a los contenidos particulares de cada época y cada situación. Como en el adagio chino, no se trata de entregar el pescado al hambriento, sino de enseñarle a pescar.

La misma idea de docente se ha reformulado. Los profesores precisan de una formación permanente para no defraudar a sus pupilos y no quedar desfasados. Esta preparación del docente debe ser una línea de tensión dirigida hacia dos objetivos; relacionados entre sí:

— una mejor inserción del joven en su entorno;

— una mejor comprensión por parte del joven de su entorno.

Participar y comprender aquello que nos rodea, para que se nos acoja felizmente y para que podamos intervenir en su perfección o cambio; pero cabe preguntarse: ¿cuál es este entorno? ¿y cómo es?

El entorno o medio de esta segunda revolución industrial o era de la computación es muy distinto del de la era agrícola o del de la era industrial, que se inicia a finales del siglo XVIII y se agota en el tercer cuarto del siglo XX. (Cuando se hace referencia a *antes* se está indicando la era industrial, y por *ahora* se entiende la era de la computación.) Este entorno se caracteriza esencialmente por dos fenómenos:

— *tecnificación*;

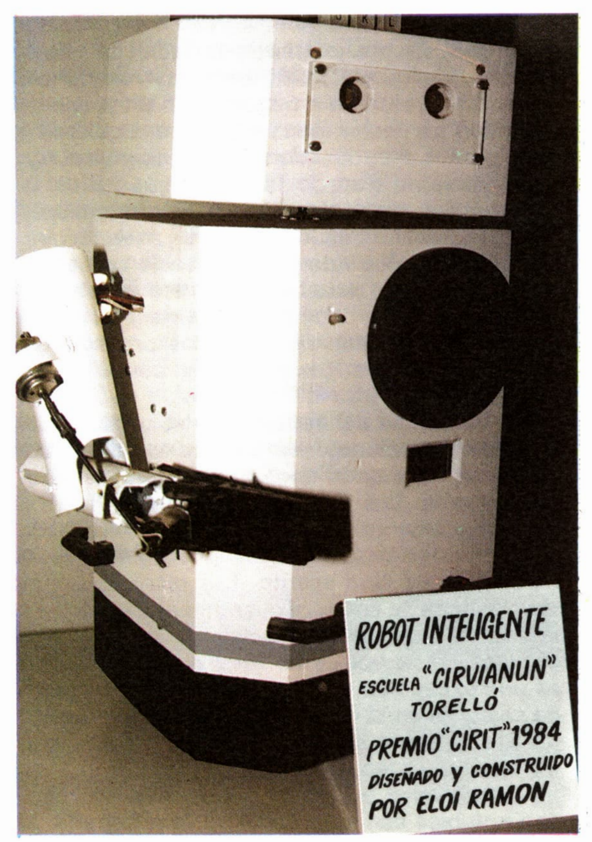
— *complejidad*.

La imbricación de los desarrollos técnicos en la vida cotidiana comienza a ser realmente dinámica y, lo que es más sorprendente, casi instantánea. Antaño, cualquier invención técnica necesitaba décadas para hallar su espacio comercial y social. Actualmente, la utilidad y los efectos de cualquier invención reciente se aprecian con rapidez inusitada. Y esta cadena de realización tecnológica comienza a influir en las relaciones entre el individuo y la sociedad, por un lado, y entre lo humano y lo material, por otro. El individuo se relaciona con sus semejantes no de forma directa, sino *tecnologizada*, y estas formas cualificadas y sofisticadas de comunicación también se interponen entre el colectivo humano y aquello que se considera como un mundo material, proporcionando a aquél una imagen concreta de éste.

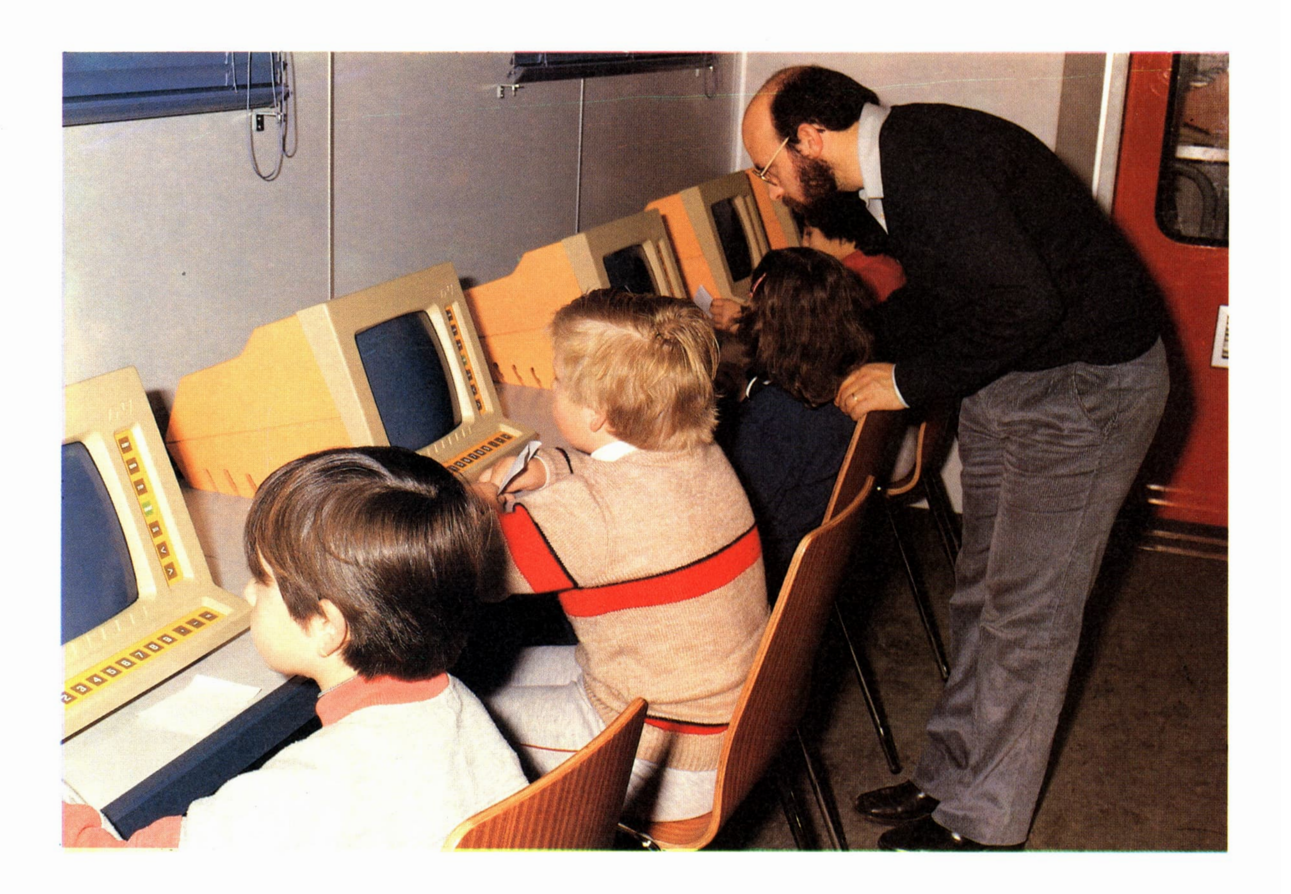
La medida de las cosas ya no está en el hombre, sino en los desarrollos del hombre. Ello no ha de llamarnos la atención ni escandalizarnos. Las invenciones técnicas no son sino un reflejo variado y complejo de su creador, el hombre.

La escuela es un recinto que generalmente ha estado apartado de los fenómenos sociales más vivos. De ahí que el contraste entre escuela y sociedad se agrande cuando esta última experimenta cambios sustanciales como los que se están viviendo en las últimas décadas de nuestro siglo. Si contemplamos el panorama social de este entorno tecnológico y complejo,

Niños utilizando una computadora: las formas de aprender deben evolucionar para aprovechar las inmensas posibilidades educativas de las computadoras.


ROBOT INTELIGENTE
ESCUELA "CIRVIANUN"
TORELLÓ
PREMIO "CIRIT" 1984
DISEÑADO Y CONSTRUIDO
POR ELOI RAMON


126 :3=
=?



Cada día más, la computación formará parte de la cultura básica que se impartirá a los niños en las escuelas. Arriba, curso de introducción a la computación.

constatamos ciertos rasgos que resultan determinantes y a la vez envidiables (todo ello visto bajo el prisma de la escuela). Éstos son:

— vitalidad;
— versatilidad;
— variedad;
— riqueza.

La función escolar consiste, por lo tanto, en que la enseñanza sea permeable a los fenómenos sociales. Ha de ser receptiva y extraordinariamente sensible respecto a lo que acontece a su alrededor y a cómo debe formar. De este modo recreará en su seno la vitalidad, versatilidad, variedad y riqueza del entorno. Este proceso ya está en marcha e introduce en la enseñanza unitaria los conocimientos tecnológicos y las formas de adquisición del conocimiento.

LOS INSTRUMENTOS DE LA ESCUELA

En ciertos aspectos se aprecia un fuerte contraste entre el medio social y el escolar. Muchos elementos se han ido creando y, paulatinamente, se han ido incorporando a la sociedad, y uno tras otro se han perdido para su uso en la escuela. La imprenta, la radio, la televisión, el vídeo, etc., son ejemplos relevantes. Apenas han entrado en el aula con un proceder didáctico particular; pensando en cualquiera de ellos parece que deberían haber encontrado un uso realmente pedagógico. La imprenta de tipos móviles procede del siglo XV. Muchos siglos han pasado desde entonces y, aun así, este dilatado período de plena vigencia tecnológica no ha servido para hallar su justo papel en la producción y reproducción de textos, tanto los de los propios alumnos como los procedentes de otras fuentes generalmente consideradas cultas.

Si la historia del ser humano es un largo camino de invenciones técnicas, su estudio debería correr paralelo al uso comprensivo de algunas de ellas. Las formas orales de transmisión del conocimiento dejaron paso al libro. Desde hace cinco siglos, el cauce privilegiado de la cultura no es sino escrito. Las técnicas y formas propias de esta cultura renacentista no han entrado plenamente en la escuela ordinaria. La producción de textos en serie por los alumnos es una excepción y una rareza. Lo que se escribe se agota en sí mismo y no puede circular entre los posibles lectores del aula, salvo como manuscritos irrepetibles.

Algo similar ocurre con otros avances, pero con notables cambios. La radio, la fotografía, la

discografía, la televisión y el vídeo, abren unos nuevos canales. Ya no se trata de lo oral frente a lo escrito, sino también de lo escrito frente a lo oral y lo visual. Es un complejo que evoluciona desde el estadio de lo que se «dice» al de lo que se «escribe»; después avanza al estadio de lo que se «dice» y lo que se «ve».

EVOLUCIÓN HISTÓRICA DE LA CULTURA

La evolución histórica de las formas culturales presenta diversas etapas. Una primera, realmente amplia, abarca formas comunicativas eminentemente orales: por más que se conocía la escritura, no se disponía de medios técnicos para la reproducción de textos. Una segunda etapa, revolucionaria para el saber, se inaugura con la invención de la imprenta. Esta segunda etapa sigue vigente, pero con apreciables alteraciones a causa del advenimiento de los medios sonoros y visuales.

Es evidente que, en lo que afecta a la educación, generalmente se ha insistido en formas de conocimiento oral y en el mero uso (no creación) de elementos o productos propios de la era de la comunicación escrita y de la era audiovisual. También es evidente que los intentos por acercar la televisión o el vídeo al medio educativo resultan poco efectivos, salvo experiencias excepcionales. La televisión llegó con veinte años de retraso a la escuela. Los responsables y educadores han querido estudiar con profundidad esos mismos medios, y el tiempo que ello les ha exigido se ha traducido en un retraso de su previsible incorporación escolar, porque su uso social se ha impuesto con rapidez, naturalidad y eficacia aplastantes.

He aquí que, en medio de esta encrucijada de rasgos de varias etapas y sin la clara determinación de un modelo tecnológico que proporciona vigor a las prácticas educativas, se abre paso una nueva etapa. Se trata de la era de la computación, en la que los procesos sociales más cualificados giran en torno a la producción, distribución, selección y almacenamiento de información. (El sentido de «información» es más amplio que el que normalmente se equipara al de noticia.) Con esta nueva era llega (no como efecto, sino como causa) una invención especialmente útil para los propósitos educativos: la computadora.

La computadora significa la reunión de todas las formas culturales que han existido y su integración potente. Recoge imagen, sonido y, en un futuro próximo, permitirá la comunicación oral entre la máquina y el hombre. Respecto a este último aspecto, existen sintetizadores de voz, pero constituyen sólo una anécdota de lo que cabe esperar.

La computadora: el nuevo libro electrónico

La radio y la televisión, comparados con la computadora, son elementos educativos blandos. La computadora constituye el libro electrónico, nuevo soporte económico, atractivo y dinámico, que supera la escisión entre palabra e imagen para proporcionar la interrelación explosiva de lo textual y lo icónico. No representa tan sólo un cambio de material en la transmisión de la información; que aparezca una pantalla y un teclado es algo secundario. Lo que aporta es un cambio formal en la adquisición de los conocimientos y de las habilidades.

La asunción plena de una cercana generación de computadora-libro de texto, con la que se pueda «hablar» en lenguaje natural, supondrá la adecuación de la escuela al nuevo paradigma cultural, a la nueva etapa del conocimiento. También supondrá un cambio profundo, cualitativo, de ritmo y de sensibilidad. Con ello es posible que se corrija de manera notable el escoramiento de la institución escolar hacia el pasado, su gusto por el recuerdo y la repetición e imitación de sí misma.

LA COMPUTADORA Y LA EDUCACIÓN

La aparición en el mercado, a finales de los años setenta, de las microcomputadoras ha significado un cambio sustancial del panorama computacional. La construcción de estos pequeños aparatos, que contrastan con los grandes equipos, es anterior a la fecha indicada; pe-

La figura muestra que la computadora constituye el punto de encuentro de todas las formas de comunicación cultural hasta ahora existentes (imagen, sonido y, en un futuro próximo, comunicación oral entre usuario y computadora).



ro este tímido tanteo de los fabricantes en un ámbito desconocido, el de los pequeños usuarios, fue perfilándose en pocos años como un éxito de gran magnitud. De ahí que los avances en la microcomputación permitieran desarrollar decididamente la línea iniciada. Actualmente, el mercado de las microcomputadoras es sumamente importante, tanto por lo que se refiere a ventas como por lo que atañe a su definitiva repercusión y aceptación sociales.

LAS MICROCOMPUTADORAS EN LAS AULAS

Las microcomputadoras encontraron aplicaciones enormemente interesantes en los negocios, como la pequeña empresa, y en la actividad profesional. No habían sido pensadas para la educación; pero, si nos remontamos algo más, varias décadas, comprobamos que las primeras computadoras tampoco habían sido diseñadas para la actividad comercial, sino únicamente para aplicaciones en el área de la investigación y en el sector bélico y militar.

Las microcomputadoras entraron en las aulas merced a la presión de padres de clase media, profesionales liberales y empleados cualificados, que tenían relación en su trabajo con el desarrollo de la computación. Éstos se apercibieron de la gran utilidad de las computadoras en su trabajo y manifestaron que, con su uso, mejoraba el aspecto mecánico y conceptual de muchas de sus tareas.

Esta sensibilización se produjo en medios familiares de zonas urbanas y altamente desarrolladas de Estados Unidos. Si bien es cierto que la historia de la computación en la educación es anterior (con la aplicación de grandes computadoras a programas universitarios), a partir de finales de los años setenta se produce el comienzo de un fenómeno que en breve resultará general: la implantación de la computación en la escuela de manera integral.

Hablar de cifras (del número de computadoras por aula, del número de programas educativos disponibles, del número de horas que habría que dedicar a la computación, etc.) será siempre un intento condenado a resultar efímero. Millones de aparatos están ya instalados y a ellos se sumarán muchos más, y, en poco tiempo, éstos serán sustituidos por otros de características que hoy son casi imposibles de imaginar.

Una revolución en las escuelas

Las escuelas no han experimentado cambios importantes en los últimos doscientos años, ni siquiera en los últimos dos mil años. Sin embargo, una persona de nuestro tiempo no se sentiría cómoda en una escuela del año dos mil. La generalización del uso de las microcomputadoras en las escuelas supondrá una revolución mucho más grande que la que produjeron la invención de la imprenta o el uso de los medios audiovisuales.

Walter Mathews

La computación en la escuela

La computación está cambiando la escuela. Las formas de aprender ya no son las mismas y aún cambiarán más. No se trata de la incorporación de un elemento instrumental de escaso radio de acción. No se trata de reproducir la organización hasta ahora conocida, ni los viejos esquemas pedagógicos en los que el profesor juega un papel central y excluyente. El cambio que se puede operar no consiste en la sustitución del profesor por la computadora, ni tampoco en el uso de la computadora como mero auxiliar. Éstos son dos extremos que no se ajustan a las capacidades de la revolución educativa. La posición aconsejable se halla en un punto medio que combina dos elementos irreductibles: el humano y el tecnológico; el profesor y la computadora.

Con tecnología o sin ella, el profesor está dejando de ser el único transmisor de conocimientos en el aula. Los medios de comunicación social compiten con él y suministran a los niños y jóvenes todo tipo de elementos informativos y formativos, con la ventaja de poseer formas sumamente atractivas, variadas y ágiles. El papel del profesor ya no es el de mediador entre los alumnos y la realidad, sino más bien el de un coordinador de las tareas escolares, un asesor, un conductor de procesos pedagógicos creativos y activos. Por su parte, la computadora puede constituir una cantera sin limitaciones para el trabajo escolar, la indagación, los tanteos y el hallazgo de respuestas o soluciones. Con ello se responde a un principio pedagógico que se impone por sí mismo: *el alumno no precisa aprender las soluciones de problemas concretos, sino la forma de resolver estos u otros problemas cualesquiera.* Lo fundamental es el procedimiento.

Las aulas cambiarán su disposición para permitir un tipo de trabajo más personalizado. Los alumnos dispondrán de fuentes documentales y de ámbitos de simulación de situaciones mediante el uso de la computadora, de los bancos de datos y de las redes telemáticas. El aspecto de un aula podrá parecerse a una biblioteca, en la que los alumnos, individualmente o por grupos, trabajarán en sus deberes, y solicitarán la ayuda del profesor cuando la necesiten.

Esta colaboración entre la computadora y el profesor permite conseguir lo mejor de cada



El aprendizaje de la lectoescritura puede facilitarse enormemente gracias a las posibilidades interactivas de las computadoras.

uno de ellos. De esta forma, el profesor no siempre es acaparado por toda la clase y puede prodigar atención personal y aportar sus indicaciones metodológicas y sus explicaciones. Las tareas mecánicas de control, evolución, elaboración material de listas y calificaciones, suministro de material informativo, recordatorio de notas, etc., todo ello puede ser realizado por la computadora, según las prescripciones del profesor. Éste queda liberado de todo aquello que no sea creativo (cuantitativamente considerable) y puede dedicar mayor tiempo y atención a sus alumnos y a la labor esencial de su profesión.

Al mismo tiempo, la computadora ofrece al alumno una ayuda inestimable. Dejando de lado las nuevas formas de operar y de pensar que ofrece y suscita, la máquina posee unas cualidades que la hacen enormemente apta para la pedagogía:

— objetividad;
— paciencia;
— capacidad de control;
— carácter didáctico.

Mediante la introducción de informaciones concretas, la programación de unos parámetros objetivos e idénticos de control y la posibilidad de repetir todos los ciclos programados hasta la saciedad, es posible encontrar en la computadora un aliado pedagógico.

Respecto a su carácter didáctico, esta expresión ha de entenderse como la capacidad de desplegar un método de acercamiento y presentación eficaz de los contenidos. Para ello cuenta con diversos recursos, a saber, la ejemplificación, la animación de figuras, el color, los gráficos, los efectos sonoros, etc. Todas estas características no son necesariamente inherentes a la computadora y deben arraigarse en un adecuado software. De ahí que, cuando se habla de la computadora, siempre se tenga presente al profesor o al técnico de programación.

La computadora no desplaza al profesor. Éste aporta al aula varias características imposibles de hallar en la máquina:

— inteligencia;
— personalidad;
— calor humano;
— creatividad.

La actividad del profesor no está sujeta a un patrón fijo, ni tan siquiera a una red de patrones. Su inteligencia le permite atisbar nuevas situaciones o detectar dificultades y soluciones no previstas. Su personalidad puede constituir un estímulo determinante y su particular planteamiento de los quehaceres motivará más que unas formas estandarizadas de presentación. Por último, el calor humano de su relación puede superar barreras que difícilmente se salvarían de otro modo; además, la creatividad e inventiva anima a aprovechar los recursos electrónicos que se ofrecen al alumno.

Con lo que antecede queda clara la defensa del binomio profesor y computadora. Es una relación necesaria en la que cada elemento aporta unos valores marcadamente poderosos; en última instancia, la computadora es la conjugación de unos dispositivos físicos y una laboriosa planificación desplegada por los profesionales de la educación.

El encuentro entre computación y educación plantea cuestiones más profundas que las meramente históricas o las relativas al diseño del aula, a las características de los elementos que actúan y al binomio computadora y profesor. Dicho encuentro plantea algo tan radical como el surgimiento de nuevos papeles, nuevas relaciones, nuevos objetivos y nuevas metodologías.

Considerar estos puntos y desarrollar respuestas y estrategias constituye un objetivo de gran importancia. Contemplado analíticamente, este objetivo iluminador consiste en responder a preguntas como las siguientes:

- ¿Quién enseña?
- ¿Qué se ha de aprender?
- ¿Quién o quiénes tienen que aprender?
- ¿Quién suministra la información?
- ¿Cómo se suministra la información?
- ¿Qué relación se establece entre los diversos elementos?
- ¿Qué metodologías cabe desplegar?
- ¿Cómo se configuran las materias o asignaturas y los cursos?
- ¿Cómo se evalúa la rentabilidad educativa y social?

Tal vez no sea posible, ni tampoco deseable, obtener unas respuestas definitivas a estas preguntas. La educación que se prevé es la educación del cambio, aquella que es dinámica y no cristaliza en formas válidas fuera de unos límites temporales reducidos. No obstante, si se contemplan preguntas o cuestiones que apuntan hacia esa dirección, se está dando un gran paso. El paso consiste en asimilar el concepto de cambio del sistema escolar.

ALFABETIZACIÓN ORDINARIA Y ALFABETIZACÍON COMPUTACIONAL

El concepto

El término «alfabetización» resulta polisémico. Responde a un sentido recto y a otro metafórico. El sentido recto habla del aprendizaje de la lectura y de la escritura. El sentido metafórico se refiere a la adquisición de unas técnicas, unas habilidades y una terminología necesarias para el uso de las fuentes de información técnicamente más cualificadas y de progresiva implantación social. El sentido expresado en primer lugar puede equipararse al concepto de alfabetización ordinaria. El segundo sentido abarca el concepto de alfabetización computacional, que es más amplio y significa el desarrollo de la capacidad de computación, que es una puesta al día de la capacidad de leer y escribir.

Dejando a un lado la polisemia arriba indicada, la palabra «alfabetización» está cargada de una fuerte ambigüedad, propia de un concepto tan relativo. Admite múltiples grados, dentro de su escala, y los más ínfimos remiten al analfabetismo funcional, esto es, a la incapacidad real de utilizar unas supuestas habilidades lectoescritoras. El sujeto lee y escribe, pero no sabe comprender un mensaje escrito ni codificar un mensaje propio de manera expresiva y eficaz.

Por tal motivo se suceden redefiniciones que hablan ya del dominio de las técnicas instrumentales básicas (como son la lectura, la escritura y el cálculo), ya de una forma de ejercer la memoria y de organizar el pensamiento. Quien pertenece a una sociedad alfabetizada no puede pensar con los esquemas de aquella otra que vivió en una sociedad de formas orales de transmisión del conocimiento.

Relación entre situación y texto

El pedagogo Pablo Freire asevera que «el acto de leer no se agota en la decodificación pura de la palabra escrita, sino que anticipa y se alarga en la comprensión del mundo». En virtud de ello, hemos de entender que la lectura del universo —de la realidad cercana y lejana a nosotros— ha de realizarse de manera previa a la lectura de palabra. Si no se comprende la realidad, no se puede dominar el lenguaje. Tal puede ser la razón de los múltiples casos de analfabetismo funcional.

Esta relación inevitable entre la situación (lo que llamamos la realidad material y cultural) y el texto evidencia doblemente la necesidad de la reforma del *curriculum* escolar y la introducción de la computadora como herramienta metodológica.

Sistemas de alfabetización				
ALFABETIZACIÓN ORDINARIA	Lectura	Escritura	Cálculo	
ALFABETIZACIÓN COMPUTACIONAL	Lectura	Escritura	Cálculo	Técnicas y habilidades relativas al uso de fuentes de información cualificadas.

La computadora relaciona la actividad escolar con el medio, con la realidad y el paradigma cultural. Pero, además, la máquina no sólo mejora y optimiza el aprendizaje, sino que, situados en el punto cero del aprendizaje, facilita la adquisición de los primeros rudimentos de la lectura y la escritura, inicia en el dominio del teclado y proporciona soltura en el uso de los recursos computacionales.

Aprender a leer con la computadora

En este último supuesto se produce el fenómeno del ingreso simultáneo en las dos galaxias: la galaxia Gutenberg y la galaxia tecnológica. De esta manera se alcanza la forma superior de alfabetización. Resulta indudable que la práctica de la alfabetización ordinaria por medio de la computadora supone en sí misma el inicio de la alfabetización cualificada.

ETAPA PREALFABÉTICA

Se han realizado experiencias solventes en este ámbito y pueden extraerse algunos parámetros valiosos. En la iniciación a la lectura y la escritura mediante la computadora de niños de corta edad se distinguen dos etapas. La primera es la *etapa prealfabética*. En ella el niño puede manejar, cuando sólo cuenta unos meses de edad, un teclado especial, compuesto de comandos y teclas de colores que producen sonidos y figuras en la pantalla. Mediante teclas de desplazamiento imprime movimientos y ubica bloques coloreados o dibujos.

Los programas que conducen al niño ágrafo, a través del alfabeto, a un universo plena-

Además de un medio que facilita su aprendizaje, para un niño de corta edad, la computadora puede constituir un objeto extraordinariamente lúdico.



El alumno frente a la computadora

El alumno aprieta el interruptor que conecta con la red y, con este gesto, abre el libro comenzando siempre con el índice. Para llegar a la lección que le interesa no se ve obligado a pasar páginas; basta con que diga cuál es o, mejor dicho, que lo escriba, e inmediatamente aparece.
El libro es a la vez libreta y pizarra, ya que permite realizar en él los ejercicios. Los resultados quedan reflejados en cada intento y son borrados posteriormente.
El libro incluye texto, imagen en movimiento y sonidos; esto es, reúne al libro audiovisual y al libro tradicional. Además, rescata para la modernidad al espejo mágico de los cuentos fantásticos con su capacidad de dialogar con su dueño, contestar o proponer acertijos, señalar el éxito o el error de las manipulaciones, sugerir repeticiones o formas de recuperación y, fundamentalmente, reflejar en su pantalla las buenas cualidades intelectivas de su usuario.
Este libro, que no está hecho de papel, sino de circuitos integrados, y que no está escrito en tinta, sino con impulsos eléctricos en virtud de un alfabeto binario, constituye la más actual metáfora de su lector. Esta máquina se erige en metáfora multiforme, dinámica y personal. Esta máquina es la computadora.
Frente al libro, que es un elemento pasivo (a pesar de las múltiples resonancias que puede suscitar), la computadora puede desempeñar el papel de «tutor» del alumno. Y éste la recibe con el aplauso de su aceptación duradera.
En esta capacidad de la máquina para ser tonta o fantásticamente «inteligente», según las habilidades del usuario, se halla la metáfora del espejo electrónico. El monitor de la computadora nos presenta la faz multiforme de su dueño: mediante una traducción analógica, los despliegues gráficos y los parpadeos textuales reflejan de forma dinámica y personal las bondades intelectivas del usuario.

mente significativo resultan generalmente eficaces a partir de los cinco años. No obstante, a partir de los tres o de los cuatro años, mediante una adecuada gradación, es posible iniciar el aprendizaje con éxito. La organización de los períodos puede ser diversa. Ello depende de las características del paquete de programas y de las condiciones ambientales. No obstante, puede simplificarse en dos fases: la analítica y la sintética.

FASE ANALÍTICA

Ante todo, una consideración fundamental: lo que mueve al niño a adquirir destreza y a reconocer el alfabeto no es el interés por la lectura, sino la curiosidad y el reto que significa dominar la máquina y el programa que se ha cargado en la memoria.

La primera fase resulta propiamente analítica, de paciente y progresiva distinción de los elementos o rasgos. El programa pretende el aprendizaje de un abecedario peculiar.

Cada tecla está programada para representar un dibujo distinto. Así, si se presiona la tecla «A» puede aparecer en pantalla la imagen de un árbol o, si se hace lo propio con la «G», la de un gato, etc. El juego al azar con las teclas y el cursor permite confeccionar cuadros de sugestivo poder plástico.

La relación entre el epígrafe de la tecla (aquello que aparece impreso en la tecla de la máquina) y el dibujo puede ser arbitraria o puede no serlo; pero, en definitiva, se ofrece una relación convencional. De esta suerte se genera una representación por orden del objeto. La letra inicial se presenta en pantalla con un carácter grueso y llamativo. Con todo ello se combina el dibujo y la grafía. Dibujo y grafía aparecen en un curioso plano de la realidad del niño, que es la pantalla.

Mediante este sistema, la tecla-letra (cada tecla suscita una imagen y una letra determinada) es la llave que franquea el acceso a la imagen. Este punto es sumamente importante y se diferencia del método tradicional, en el que la imagen conduce a la letra. En todo caso, mediante la computadora y este programa se establece un corredor de doble sentido, que desemboca, en cada recorrido lineal, en la atención a las imágenes de interés y en la retención de las letras iniciales (alfabeto). Se parte de la tecla para acceder a la imagen. Y la imagen nos remite y asocia con la letra en la pantalla.

Este aprendizaje al azar implica el trasiego repetido de todas las posibilidades. Todas las teclas son accionadas repetidamente y memorizadas convenientemente. Esta tarea se cumple de manera lúdica, voluntaria y espontánea. El alumno controla su juego y decide cuándo y lo que quiere aprender. Sólo recibe ayuda cuando la solicita, porque no es preciso más.

FASE SINTÉTICA

Con un cambio de programa se accede a la segunda fase, que se ajusta a una función sintética. En esta fase se recogen los progresos realizados y se penetra en la dimensión significativa de la palabra.

La pantalla presenta un paisaje compuesto de varias secciones figurativas horizontales o verticales. Por ejemplo, puede dividirse en los siguientes estratos, dispuestos de arriba a abajo: «cielo», «prado», «camino» y «río».

El niño dispone de unas fichas auxiliares en las que aparecen escritos, con caracteres grandes y claros (idénticos a los del teclado), nombres de cosas, como «coche», «sol», «avión», «árbol», «nube», «barca», «perro»... Cuando el niño teclea todo el nombre que aparece en la ficha, cuyo significado desconoce, se representa en la pantalla su figura. El niño relaciona el dibujo con la representación ortográfica, con lo que la asociación es aún mayor, pues aparece en tres elementos: imagen acústica, imagen gráfica y realidad.

El juego no resulta mecánico. Con teclas de desplazamiento en cualquier dirección de giro y de velocidad, se puede proporcionar vida al paisaje, ubicar el dibujo a placer y componer una historia imaginativa con la adición de otros dibujos.

El resultado de toda esta actividad, que se dispone a lo largo de cierto tiempo y no se propone como un objetivo en sí, es que el alumno aprende las palabras; es decir:

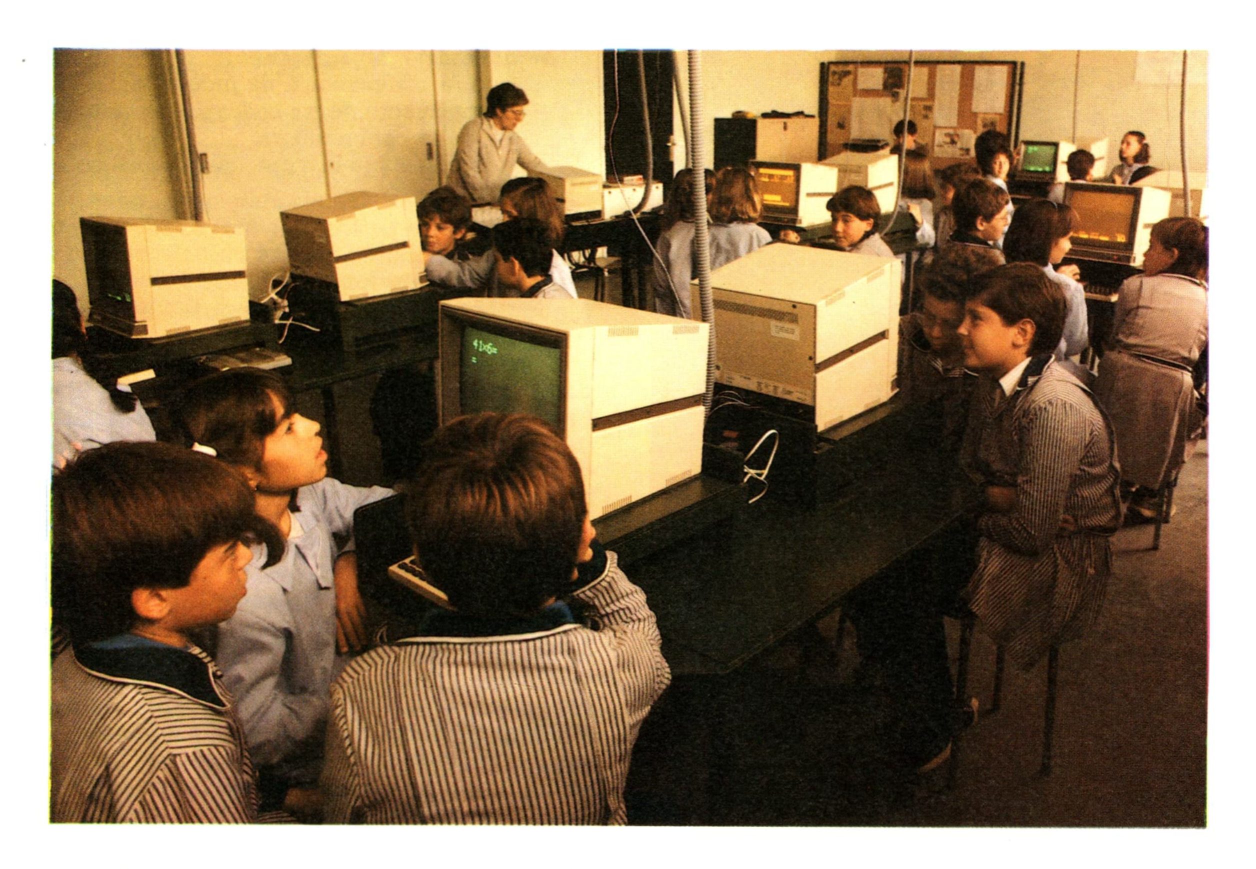
- reconoce el vínculo convencional entre la grafía y el concepto;
- pronuncia las palabras;
- teclea las palabras de memoria.

Lo fundamental de todo ello es que, de manera simultánea, se adquiere el concepto de palabra y su materialidad, mediante la memorización de un rico y expansivo bagaje.

También cabe la posibilidad de que el niño dibuje la imagen de sus propios objetos, aquellos que no le han sido suministrados por el programa. A continuación los «bautiza», les da nombre. La computadora registra la imagen en la memoria mediante el nombre indicado. Con el teclado de las letras de la nueva palabra el niño podrá rescatar la imagen cuantas veces lo desee.

Finalmente, el niño reconoce las palabras de las tarjetas si aparecen con otra grafía. Reconoce, también, las letras y las asocia con las imágenes acústicas y conceptuales.

Merced a este juego educativo, que no presenta grandes diferencias frente a los juegos que se pueden hallar en los catálogos de software disponibles, el niño descifra el significado de las palabras al representarse su imagen en la pantalla. No es preciso esperar a que se desarrolle una cierta capacidad abstracta para comprender; el alumno puede incluir en su vocabu-



El trabajo en grupos, por su carácter interactivo, parece un medio adecuado para el aprendizaje de las matemáticas, como pone de relieve la fotografía superior.

lario términos cuyo uso se encuentra muy alejado de su medio natural.

Leer y escribir con la computadora

Como sea que la computadora «entiende» las palabras que el niño no comprende, éste puede aventurarse a investigar por su propia cuenta. Fortalece la habilidad conquistada para leer y escribir y aumenta su tesoro lingüístico, su diccionario propio, su mundo referencial. A la vez, interioriza el poder general de las palabras para dirigir o dominar a la computadora. Esto último es vital. Induce la idea de que las palabras son instrucciones, son el motor de la acción. La computadora actúa espoleada por una palabra tecleada. En los otros ámbitos de la realidad, los que le rodean también se mueven a instancias de algunas palabras a las que se muestran más sensibles.

LA ALFABETIZACIÓN COMPUTACIONAL

No basta con usar los recursos computacionales para el aprendizaje de la lectura y la escritura (en la iniciación a la cultura alfabética) ni para escribir y leer correctamente, según los sistemas de procesamiento de textos. La computación se erige en cultura, cultura básica que impregna formas de saber y de hacer. La expresión «alfabetización computacional» describe el fenómeno —y la necesidad— del adiestramiento en las técnicas y aplicaciones computacionales. (En inglés, el concepto se conoce como «Computer literacy».) Estos nuevos conocimientos y destrezas se imparten en la escuela y forman una parte significativa de los planes de estudio. Éste es un aspecto de la introducción de las nuevas tecnologías en la educación. Otro mayor y más ambicioso es el de la aplicación sistemática e integral de los nuevos recursos a las nuevas materias.

Evidentemente, no es lo mismo enseñar computación que utilizar la computación para optimizar la enseñanza de las distintas materias. Las dos actividades no se excluyen; se complementan y potencian. En la primera, la computación es un objeto de estudio. La segunda consiste en el despliegue de recursos didácticos de especial incidencia.

La alfabetización computacional afecta, en general, a todas las personas; si bien los escolares se hallan en mejor situación de hacerse con la nueva gramática de la cultura tecnológica. Ya sea de forma plena e intensiva o bien de

El aprendizaje como experiencia gratificante

El lenguaje es pura acción. He aquí el elemento fundamental que proporciona el medio electrónico para el cambio metodológico y de motivación. Se altera, de forma positiva, la forma de trabajar y los estímulos que mueven al niño a aprender con mayor ilusión.
Si los métodos de alfabetización corrientes se resienten de la servidumbre de pasividad que se impone al alumno, la computadora ofrece la transformación del aprendizaje de la escritura en una experiencia tan gratificante, por inmediata y autónoma, como lo es la del aprendizaje del lenguaje oral. En primer lugar, la lectura y la escritura quedan identificadas, sin que aparezca ningún corte o discontinuidad entre ellas. Ello contrasta notablemente con la disociación que aparece en los métodos tradicionales; con éstos, primero se ha de aprender penosamente a escribir y, tras un período de «incubación», se accede al nivel superior de la lectura. En segundo lugar, mediante la computadora, cualquier palabra que se pueda leer puede ser utilizada si se introduce en el aparato; en cualquier momento se puede extraer su carga significativa.
Que aprender a escribir sea una experiencia tan agradable como aprender a hablar significa la obtención de una satisfacción inmediata. Se posee un poder instantáneo sobre el entorno. Por de pronto, la escritura entraña el dominio de la máquina, y el dominio de la máquina significa el dominio de la escritura y de las formas de comunicación de la realidad circundante.

manera extensiva y diluida, niños y adultos comienzan a recibir «lecciones» acerca de esta temática y a ser cualificadamente alfabetizados, es decir, alfabetizados computacionalmente.

¿Cuáles son los elementos que configuran la alfabetización computacional? Ello depende de dos variables: la madurez social respecto a este fenómeno y los progresos de la propia ciencia computacional.

La configuración de los contenidos reales ha de venir determinada por los elementos fundamentales que se enuncian:

— comprensión de los conceptos fundamentales sobre la computadora;
— utilización del software;
— lectura comprensiva de programas;
— conocimiento de las aplicaciones generales de la computadora;
— reflexión sobre la repercusion social de la computación.

Comprensión de los conceptos fundamentales sobre la computadora

Interesa conocer el concepto de computadora, las partes físicas que la componen y las funciones que éstas realizan. Este despliegue de la estructura de la máquina requiere explicaciones y comprobaciones materiales de sus elementos.

El concepto de computadora aglutina los rasgos de máquina de propósito general, de procesamiento de información, de realización de tareas mediante —y sólo mediante— la programación, de capacidad repetitiva y, finalmente, de aplicación extremadamente diversa.

La adquisición del rico concepto de computadora —que se aleja de aquello que comúnmente se entiende por una máquina— puede lograrse beneficiosamente con problemas, situaciones y analogías que ofrezcan una más profunda y multiforme representación mental de la realidad computacional. Así mismo, para conocer las partes y el funcionamiento de la computadora es conveniente *ver* y *tocar* los elementos de la configuración física del aparato; verlo por dentro. Para todo ello existen técnicas pedagógicas eficaces que combinan los usos explicativos y las formas activas y lúdicas, especialmente la dramatización y los trabajos plásticos.

Utilización del software

La continuación del apartado anterior consiste en el uso de programas de computadora. Con ello se consigue un doble objetivo:

— materializar, concretar, la potencialidad de la máquina en una tarea plena y determinada;
— desarrollar la capacidad de manejo de la computadora y los periféricos, y dar cabal cumplimiento al uso del programa.

A su vez la actualización de las capacidades de la computadora y su manejo con soltura operan un doble efecto. Por un lado, ofrece un micromundo o campo acotado, pero rico e intenso, en el que se descubre la máquina; la ejemplificación proporciona marcos de referencia muy precisos. El segundo efecto no es otro que el dominio real de los resortes mecánicos del aparato, las teclas de función y el resto del teclado, los periféricos y los comandos básicos, la comprensión de las instrucciones y el diálogo con la computadora.

Lectura comprensiva de programas

El punto tercero, relativo a la lectura comprensiva de programas, suscita un adjetivo que puede tomarse en sentido estricto o de forma extensiva. El epígrafe enuncia su sentido estricto o breve. La forma extensiva hace referencia a la capacidad de escribir programas.

El conocimiento de la sintaxis de la programación computacional no es algo superficial. Entender, como quien lee, los rudimentos de la programación, las partes estructuradas de un programa, el papel de las instrucciones principales, tiene su interés.

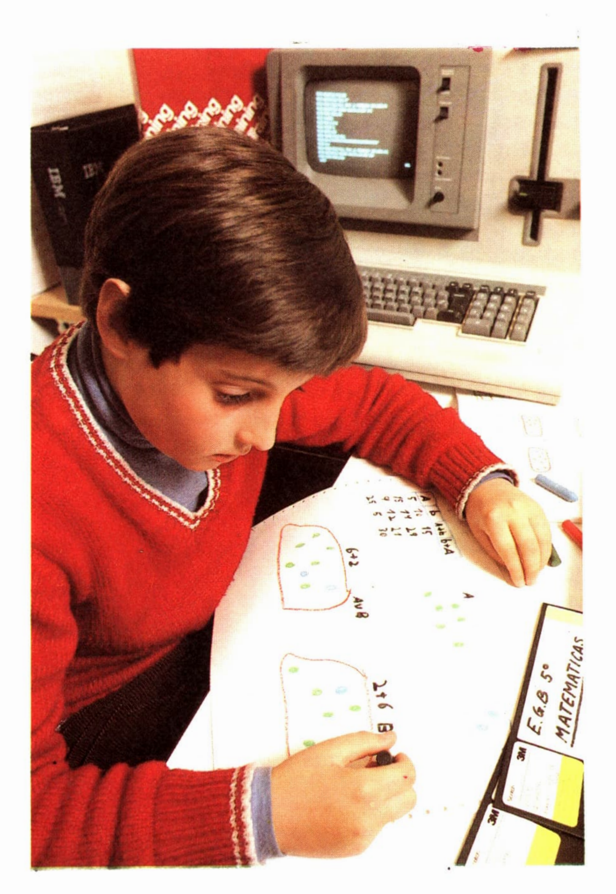
Esta lectura y comprensión puede prolongarse hasta llegar a la modificación de programas y la evaluación técnica de los mismos. Con ello se llega al conocimiento de algún lenguaje de programación. Sobre este punto se acostumbra a hacer mucho énfasis —quizá excesivo—, en el sentido de que la alfabetización computacional equivale al dominio de uno o varios lenguajes propios. Expresada de manera tan tajante, esta convención resulta muy criticable.

No cabe duda de que la escritura en BASIC, LOGO, PASCAL o cualquier otro lenguaje, no es una capacidad desdeñable. La escritura o redacción computacional implica dominar una terminología específica, conocer las particularidades del hardware y, sobre todo, el desarrollo de algoritmos y procedimientos. Esto último consiste en la disposición ordenada de un conjunto de instrucciones para la realización de una tarea. El concepto de procedimiento es más general que el de algoritmo, ya que el primero se aplica a cualquier actividad en general, mientras que el segundo se refiere al conjunto de pasos ordenados secuencialmente para resolver un problema matemático.

La estructuración de los procedimientos en un lenguaje significa un ejercicio proteico para la formalización del pensamiento. Como todo, es una tarea a la que no están llamadas todas las personas. No es tan sólo una cuestión de condiciones personales. No siempre compensa un esfuerzo tan considerable, máxime si se entiende que el desarrollo de la programación corresponde a los profesionales y que el verdadero poder de la computación radica en su uso fácil y satisfactorio.

Conocimiento de las aplicaciones generales de la computadora

El conocimiento de las aplicaciones generales de la computadora se sitúa mucho más allá del uso de simples programas. Existen aplicaciones corrientes que son en sí auténticas modalidades básicas. Es conveniente tener acceso a ellas y presentar meridianamente su carácter genérico. Entre otras, he aquí las aplicaciones



El control de aciertos y errores que permite el aprendizaje con computadora resulta especialmente útil para el aprendizaje de las matemáticas.

que pueden considerarse como más características de la tipología:

- tratamiento de textos;
- hojas electrónicas;
- archivo;
- contabilidad;
- gestión empresarial;
- contro! robótico;
- EAO (enseñanza asistida por ordenador o computadora);
- diseño industrial.

No es preciso —ni posible, en algunos casos— que la presentación de estas aplicaciones se haga en el aula o en un lugar que reúna condiciones. La visita a centros especializados es recomendable. En este último caso, la información que se pretende extraer resulta mayor y más matizada. Por añadidura, hay que considerar el intercambio de información oral y documental con los responsables y operarios del centro mediante la cumplimentación de un cuestionario o la espontánea realización de preguntas. La posterior confrontación de los datos obtenidos permite depurar errores y superficialidades y facilita la redacción de unas conclusiones comprensivas.

La consulta o la elaboración de gráficos que reflejan la cadena de operaciones de los sistemas resulta, de igual modo, sumamente didáctica. En el gráfico cabe distinguir los pasos rutinarios, las tomas de decisión, las variaciones y las anomalías previstas. Finalmente, la consideración de lo que representa la realización automatizada de las tareas frente a los sistemas manuales o semiautomáticos de otros tiempos es un aspecto que merece especial atención. Por supuesto, como ningún sistema reúne todas las ventajas, resulta instructivo contemplar los pros y los contras de las modalidades de uso computacional presentadas.

Las indicaciones anteriores nos remiten a unas condiciones generales para asegurar la utilidad didáctica del trabajo, que se resumen de hecho en estas tres:

- Adecuar la dificultad o complejidad de las aplicaciones computacionales a la edad e intereses de los alumnos. Para ello hay que graduar los ámbitos y las formas de acercamiento al objeto de estudio.
- Proponer o incluir la elaboración de un cuestionario exhaustivo que contemple algunos de estos puntos:
  - función que cumple la aplicación estudiada;
  - características de la computadora y tipos de periféricos;
  - lugar donde se utiliza la computadora;
  - personas que la utilizan;
  - tipo de información que se suministra al aparato;
  - proceso que sigue la información;
  - problemas y errores que pueden llegar a presentarse;
  - comparación del sistema computacional con otros sistemas tradicionales.
- Considerar el conocimiento de las aplicaciones de la computación no como un fin, sino como un medio. Lo fundamental no es saber qué es una «hoja electrónica», por ejemplo, sino saber utilizarla con algún objetivo. Por consiguiente, lo que interesa es llenar de contenido un canal operativo como es cualquier sistema de computadora.

Reflexión sobre la repercusión social de la computación

El estudio de los elementos técnicos desemboca en la reflexión acerca de los efectos sociales de las computadoras y los procesos telemáticos. Conviene, en primer lugar, acercarse a este tema y, en segundo lugar, hacerlo sin una opinión preestablecida o prejuiciosa.

El planteamiento de la cuestión ya implica la aceptación de que la tecnología está operando cambios que hay que analizar y calificar. El acercamiento exento de una aceptación o un rechazo absolutos es la segunda condición para desarrollar el sentido crítico.

La repercusión social es tan obvia que resulta más fácil tratarla por sectores: laboral, socioeconómico, personal. También, en cada uno de ellos, se puede elaborar una relación de ventajas y desventajas o efectos perjudiciales.

- Sobre el sector laboral se pueden tratar los efectos de empleo o desempleo, las nuevas profesiones, la actualización o el reciclaje profesional, etcétera.
- El ámbito socioeconómico puede suscitar el debate acerca de la dependencia o liberación que introduce el uso de las técnicas y medios de computación, el cambio de las formas de producción, el aumento del sector económico de la información o cuaternario, el tipo de educación, etcétera.
- Finalmente, se puede tratar todo cuanto atañe a la persona: nuevos medios para la intercomunicación, la intimidad en peligro, nuevas formas de aprender y divertirse, etcétera.

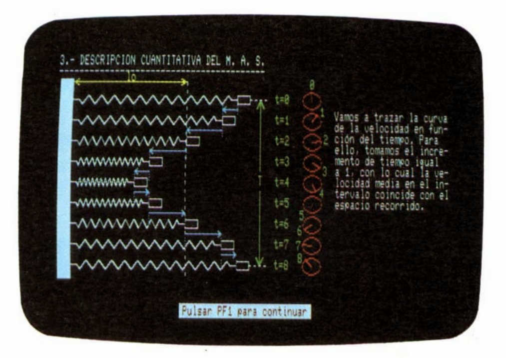
ALFABETO TECNOLÓGICO Y CRITERIO CONCEPTUAL: ALFABETIZACIÓN FUNCIONAL

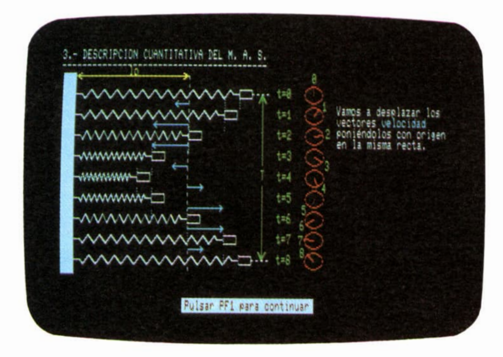
El concepto que se propone aquí de alfabetización computacional es deliberadamente amplio. Comprende la adquisición de técnicas y habilidades elementales, que constituyen algo así como el «alfabeto tecnológico», y también la maduración de un espíritu crítico. Por consiguiente, se contemplan dos aspectos: el que literalmente se entiende por la «alfabetización computacional» o *técnico* y un segundo aspecto de tipo *conceptual*.

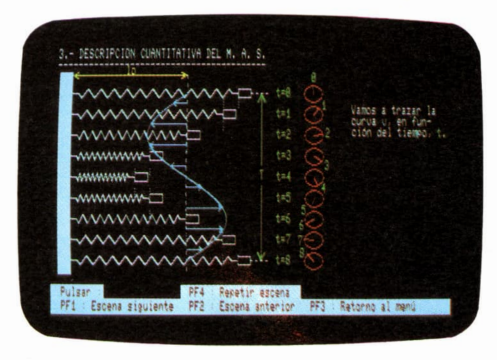
La imagen del «alfabeto tecnológico» nos remite a un código comunicativo más complejo que el estrictamente alfabético, pues integra a éste y otros elementos funcionales y tecnológicos. Con este «alfabeto tecnológico», la persona es capaz de codificar y decodificar los mensajes de una manera cualificada y tecnológica, tal como corresponde a la era de la computación en la que se entra; pero no puede comprender verdaderamente aquello que lee o escribe computacionalmente si no posee un criterio conceptual superior que le permite relacionar el texto con la realidad. Este criterio conceptual se relaciona con el sentido crítico y la reflexión acerca del modelo cultural que el cambio tecnológico aporta e impone. El criterio conceptual participa del paradigma cultural.

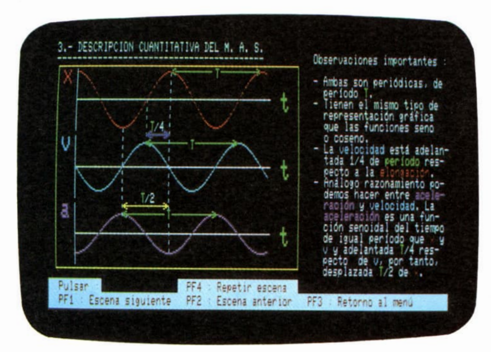
Una alfabetización computacional restringida o estrecha equivale a un analfabetismo funcional, a la incapacidad de comprensión real, a pesar de que las apariencias indiquen lo contrario. Una formación unívocamente técnica convierte a la persona en un usuario que tan sólo es capaz de consumir los medios y los servicios

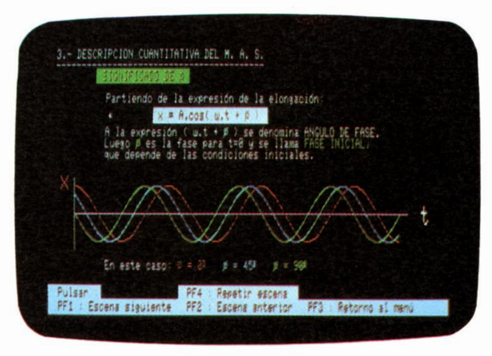


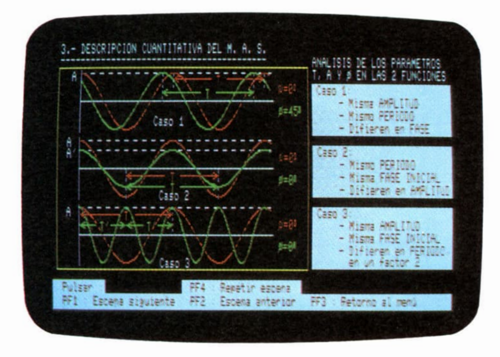


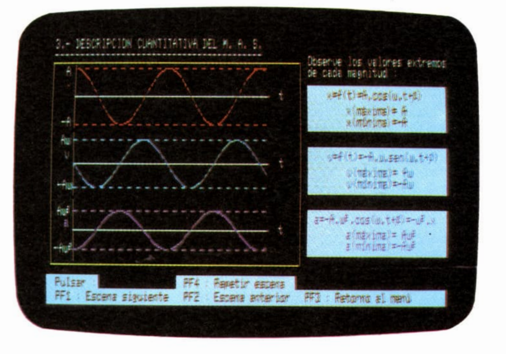












Gráficos correspondientes a algunas secuencias sucesivas de un ejercicio sobre el Movimiento Armónico Simple realizado por estudiantes de bachillerato, según un programa escrito por investigadores del Centro de Cálculo de la Universidad Politécnica de Barcelona. La posibilidad que ofrecen las computadoras para la simulación de procesos físicos ha sido utilizada en este caso para simular con fines pedagógicos un fenómeno físico sencillo, pero, en realidad, su campo de aplicación es mucho más amplio. Las computadoras se emplean cada vez más para el estudio de procesos físicos que, por su complejidad, no pueden ser formalizados matemáticamente y para la realización de experimentos que permitan confirmar o refutar modelos teóricos: las condiciones de temperatura en el interior de una estrella, por ejemplo, no pueden ser reproducidas en la Tierra, pero es posible simularlas en un programa, pudiéndose así estudiar directamente.

Alfabetización funcional

La alfabetización funcional no se reduce a la mera capacidad de leer y escribir. De manera previa y posterior se exige la lectura de la realidad, que produce las formas comunicativas y la comprensión del mundo. No se puede leer y escribir funcionalmente si no se participa de forma activa del paradigma cultural. A éste pertenecen las nuevas tecnologías y la computación.

La fotografía superior muestra unos niños practicando su aprendizaje gramatical mediante el uso del lenguaje LOGO, el cual, según los pedagogos especializados en Enseñanza Asistida por Computadora (EAC), ofrece grandes posibilidades por su enorme capacidad interactiva. El dibujo inferior pone de relieve la estrecha relación existente entre alfabetización funcional y comprensión de la realidad.

computacionales y, probablemente, de convertirse en víctima propicia. Por consiguiente, la alfabetización funcional exige una asimilación y unos hábitos racionales bien asentados. De otro modo, se puede cerrar el acceso maduro a la era de la comunicación e incurrir en un consumismo paralelo a la banalización pedagógica.



IMAGEN Y TEXTO

Los nuevos medios tecnológicos no sólo apuntan un uso más potente de la escritura, sino que relacionan la escritura y la imagen, y producen un resultado comunicativo extraordinariamente feliz. De esta suerte, se resuelve un divorcio histórico entre la palabra y la imagen, entre lo textual y lo gráfico, y se valora a la imagen con más altos criterios en los procesos de aprendizaje y expresión.

La computadora es un instrumento educativamente denso o duro. Esto quiere decir que no se acomoda al sistema sin generar una cadena de cambios que afecta a la lógica conceptual, al tipo de pedagogía y al lenguaje.

Junto al cambio de actitudes pedagógicas que exige la computadora, como es el abandono de las formas magistrales o autoritarias y la aparición de lo activo, lo independiente y personal, se produce la adaptación a una nueva sensibilidad. Esta sensibilidad no afecta meramente a una dimensión estética; resulta global y afecta a la totalidad del conocimiento.

Parece evidente la pérdida de la sintaxis tradicional. Antes se decía de alguien que hablaba muy bien, que hablaba como un libro abierto. Ahora, por el contrario, se ha de poner el énfasis en la recuperación del ver y oír; también, de la expresión gráfica. Las exposiciones generalmente están apoyadas en imágenes y en proyecciones. Ésta es la sintaxis del *pictograma*, la que comunica mediante la palabra y la imagen.

Esta forma inédita de percibir y pensar, que, por supuesto, se instala desplazando a otra forma, aporta como rico presente el aprecio intelectual de la imagen. Así alcanza la formalización. Si antes la imagen se entendía como una vulgarización del conocimiento, ahora adquiere un valor abstracto, merced a la esquematización y la simbolización.

Lenguaje e imagen se asocian en un proceso de formalización, que presenta dos polos:

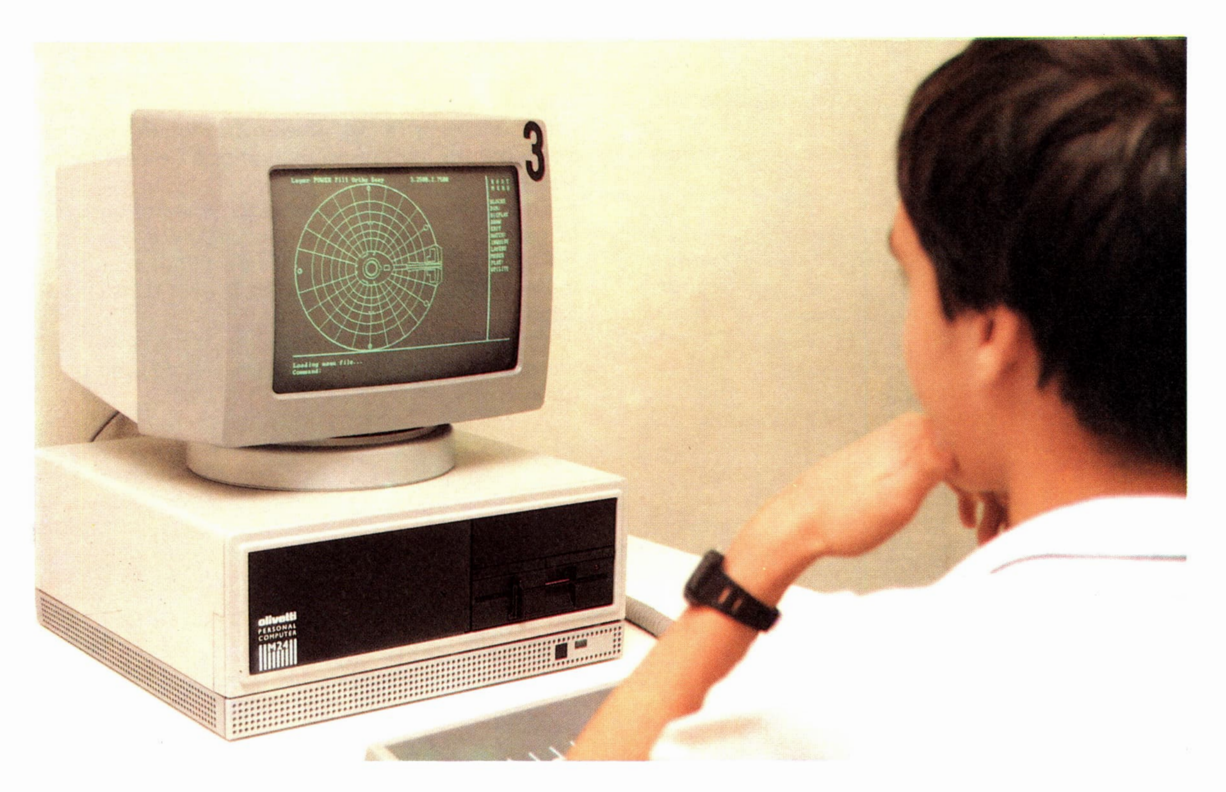
— el de la programación;
— el de la actualización de las instrucciones.

En virtud de ello, la imagen (por medio de gráficos, simulaciones, etc.,) se presenta como forma sustantiva de representación y fuente de transformación de los fundamentos de la vida mental.

Esta cultura que se impone de la imagen y la representación trae consigo la potente asociación de lenguaje e imagen. Con ello se reúne y conjuga lo abstracto y lo concreto, lo conceptual y lo visual.

Si lo icónico conforma una disposición orgánica e intuitiva del saber de gran poder sugestivo, la integración de texto e imagen proporciona una potencialidad inédita y constituye un aspecto de la mencionada nueva alfabetización.

Abajo, un adolescente utiliza un programa de familiarización del texto con la imagen en una microteca pública. Este tipo de programas pueden contribuir enormemente a desarrollar una nueva sensibilidad hacia el papel de la imagen en el conocimiento.



Programas educativos

CRITERIO PARA CONOCER LOS PROGRAMAS

La aplicación de la computación a tareas educativas se descubre tardíamente. Cuando el campo de las computadoras se expande más allá de la gestión empresarial y de la investigación, se abre paso la idea de que la educación puede resultar favorecida con el uso de estos elementos auxiliares. Aparece el concepto de *instrucción asistida por computadora* (CAI). Esto ocurría durante la tercera generación, que se extiende entre los años 1965 y 1970.

La ampliación del campo de aplicaciones de la computadora y su entrada en el área de la educación se ha operado, en esta generación de gran importancia, merced a singulares novedades técnicas. Se produce la miniaturización de los elementos electrónicos y su reunión en el circuito integrado. Los más evidentes se cifran en la reducción del volumen y consumo de los aparatos, el aumento de la fiabilidad y la renovación de periféricos.

Pero aquellos efectos que han favorecido más claramente el descubrimiento de nuevos horizontes son los siguientes:

- la multiprogramación;
- la generalización de los lenguajes de alto nivel;
- la compatibilidad y el diseño de sistemas de hardware;
- el teleproceso;
- el trabajo a tiempo compartido.

Por simples que parezcan estos elementos en esta rápida enunciación, tienen una importancia capital. Han tendido el puente de la ciudadela computacional hacia el país del gran público. El encuentro entre computación y educación que se ha producido en estos años se inició, por razones obvias, de manera reducida y titubeante. Las computadoras aún resultaban grandes, costosas y complejas, por lo que su aplicación en estos tiempos tempranos se dirigía, selectivamente, a centros universitarios y de formación profesional cualificada.

La experiencia que se ha ido adquiriendo y los grandes cambios tecnológicos han desembocado en la generalización del uso de computadoras en todos los niveles del sistema escolar. Esto se produce a partir de los años ochenta, bien entrada ya la cuarta generación.

Efectivamente, la década de los años ochenta marca un hito en la profundización de la penetración tecnológica en la escuela. La microcomputadora y el computador personal llegan a las aulas y se instalan pupitres para el uso amplio de los alumnos. Son máquinas pequeñas, baratas y muy capaces, que hacen posible la descentralización de tareas y que operan con lenguajes accesibles o, incluso, lenguajes especialmente diseñados para la escuela, como es el caso del LOGO. Y con ello llega la producción de un nutrido conjunto de programas educativos y la aplicación del concepto de instrucción computacional al de *enseñanza asistida por computadora* (enseñanza, y no instrucción).

Comprar programas y fruta

En el mercado existen decenas de miles de tiendas de material computacional donde se encuentran programas educativos. Éstos cubren la totalidad de los niveles y una amplia gama de materias escolares. Esta oferta tan extensa exige del usuario el dominio de ciertos elementos de juicio.

- En primer lugar, es imprescindible reconocer los diferentes tipos de programas según su metodología y objetivos. Se trata de distinguir varios tipos de programas y así clasificar el voluminoso conjunto en grupos homogéneos. Evidentemente, cada tipo responde a una forma de trabajo distinto.
- En segundo lugar, el usuario necesita conocer la calidad del programa que adquiere o utiliza.
- Por consiguiente, se trata de conocer la tipología de los programas y su buena o mala calidad.
- Es el mismo caso de la persona que, por ejemplo, acude a una frutería. Para comprar, debe distinguir dos cosas:
  - *El tipo de fruta que se quiere consumir*. La fruta se escoge según los gustos y la forma de preparación o consumo que se desea. Por poner un ejemplo, el plátano y el melón tienen distinto sabor y se toman de diferente manera. Y no es lo mismo escoger plátano que melón si se quiere un plato de fruta con jamón.

Arriba, clase de matemáticas en una aula especialmente diseñada para la EAC. Insertada, puntuación de los ejercicios realizados por tres alumnos e instrucciones para la corrección del juego y su continuación. Abajo, centro de enseñanza de computación de usuarios adultos para la aplicación de sistemas computacionales específicos.

Primer ejemplo: Ejercicio de léxico y memorización

En el monitor de la computadora aparece una lista de vocablos. El número de éstos es variable, según el nivel de dificultad de que se trate. Después de unos instantes, con el tiempo suficiente para que los haya leído el usuario, se borra lo escrito en la pantalla y se produce un breve tiempo de espera. Inmediatamente, la computadora avisa al alumno para que actúe. Dispone de un tiempo prefijado para escribir el máximo número de palabras que recuerde de las que antes ha leído. Con este ejercicio, el alumno controla su capacidad de memorización léxica y, a la vez, la practica para mejorarla.

• Variante primera. De entre las muchas variantes que se pueden introducir —sin considerar ni el número de términos ni la dificultad de éstos—, está la de ofrecer un ejercicio intermedio: probar la retentiva mediante un juego. La computadora ofrece al usuario un cierto número de palabras. Éste debe ir contestando «sí» o «no» según crea recordar si la palabra en cuestión aparecía o no en la lista previa. Ello facilita la tarea de memorización de un solo golpe de todos los términos y anima a perseverar para conseguir un pleno de aciertos

• Variante segunda. La atención a la corrección ortográfica no es menor en este ejercicio. En niveles inferiores, en que la ortografía es una materia importante por sí misma, el juego puede consistir en que el alumno memorice la palabra que se le presenta, retenga su figura gráfica y la reproduzca mediante el teclado cuando lo requiera la computadora. Cada sesión puede estar dedicada a la ortografía de determinadas letras y sonidos. Los ejercicios de recuperación son evaluados estadísticamente por la computadora. La participación por equipos o simplemente a base de dos «contrincantes» convierte el trabajo en un juego de percepción, memoria y rapidez.

Segundo ejemplo: Ejercicio de cálculo

En la pantalla, con números grandes, aparecen dos cantidades. Se trata de restar una de la otra. La solución que proporciona el alumno se escribe perfectamente alineada; pero no es

— *La calidad de la fruta*. En este aspecto, se atiende al grado de maduración de la fruta, la excelencia de la variedad y el lugar de origen, el aspecto y el calibre de las piezas, etcétera.

Si bien resulta impensable que una persona acuda a una frutería y pida dos kilos de fruta, sin especificar ni importarle el tipo de fruta y su calidad, no es tan raro que eso ocurra con algunos productos de la computación, lo cual es un absurdo. Puede ocurrir que la suerte depare al comprador inadvertido unos buenos programas para la computadora de su hijo o de la escuela; pero la suerte no siempre viene al encuentro, ni debemos confiarlo todo a ella.

Según los más prestigiosos especialistas, tan sólo un 5 % de los programas educativos ofrecen una calidad excelente. Y, aun poseyendo buenos programas, de poco servirán si no responden al tipo de trabajo o metodología que se desea o se puede aplicar. Sería el mismo caso absurdo de alguien que ha adquirido un magnífico destornillador para utilizarlo como martillo; necesariamente, esta persona mostrará su descontento con el rendimiento del instrumento.

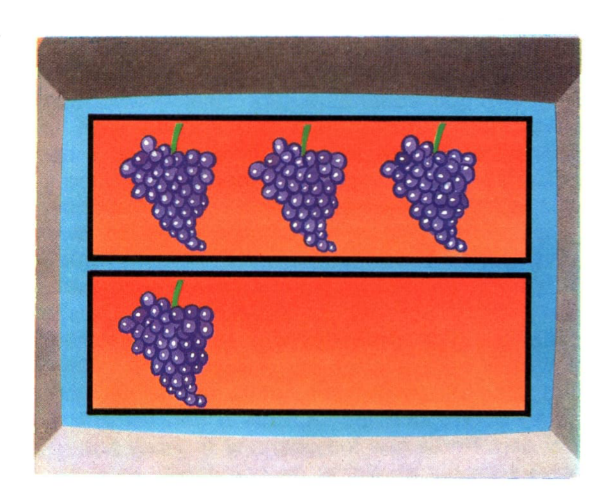
Tipología de programas

En el capítulo anterior se han presentado diversas formas de trabajo computacional en un medio educativo. Aquello que se conoce como alfabetización computacional significa la preparación elemental para la adquisición de las técnicas y la nueva tecnología de consulta y elaboración de información. La profundización en esta competencia técnica que debe tener o, en su defecto, adquirir la persona alfabetizada se extiende al uso de procesos tan elementales como el tratamiento y edición de textos y el manejo de información almacenada en soporte electrónico. A ello hay que añadir la capacidad de comprender la forma de operar con un programa sencillo y llevarlo a su cumplimiento.

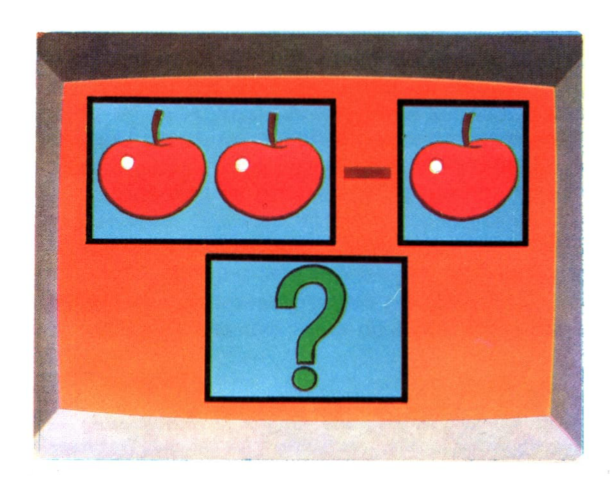
Estas aplicaciones instrumentales complementan y también se integran en los diferentes tipos de metodologías educativas. La tipología básica puede desgranarse en cinco apartados:

- Reforzamiento de estructuras.
- Simulación.
- Estructuración de problemas.
- Manejo de información.
- Control tecnológico y robótica.

El ámbito de estos grupos no es equiparable. Como se verá, los hay que ocupan una dimensión muy considerable frente a otros más modestos y también otros que están constituidos de una variedad de aspectos. No aparece explícitamente la denominación de *juegos didácticos*, porque la estrategia del juego se diluye dentro de las categorías indicadas.



Aprender jugando: Un niño de tres o cuatro años puede fácilmente empezar a desarrollar su capacidad de cálculo a través de juegos tal como los que se le proponen en los dos rectángulos que aparecen en los dos dibujos.



REFORZAMIENTO DE ESTRUCTURAS

El concepto de *reforzamiento de estructuras* reúne en su seno un conjunto de prácticas educativas sumamente ricas y efectivas. Son variadas y los progresos que con su utilización pueden experimentar los alumnos se aprecian con rapidez. Los programas de computadora que los desarrollan consiguen, de esta manera, efectos puntuales y rápidos, que luego se amplían y profundizan con otros de mayor complejidad en los grados superiores a que accede normalmente el alumno.

La exposición de varios ejemplos reales resulta el medio más claro para mostrar algunos aspectos de la gama de programas de reforzamiento de estructuras.

La inmensa mayoría de programas o software educativo que se fabrica y comercializa puede encuadrarse dentro del grupo de actividades de reforzamiento de estructuras. A la eficacia a corto término apuntada anteriormente, se suma la relativa facilidad de su creación y su manejo. La excelente aceptación del público también influye mucho.

Lo breve y puntual atrae más fácilmente la atención y el interés. Ello tiene una desventaja y una ventaja evidentes. La desventaja se resume en la posible superficialidad de la tarea educativa si estos programas no se integran en un trabajo amplio del alumno y de la clase.

correcta y la computadora «protesta» con un ruido de desaprobación y con la imagen de un monigote que pone cara triste. La máquina ofrece una segunda oportunidad para resolver satisfactoriamente la operación.
El alumno lo intenta de nuevo y tampoco acierta. Entonces el programa no insiste por el mismo camino. Simplifica el problema y guía paso a paso en las operaciones que el niño debe realizar. Si esta segunda vía es satisfactoria, se vuelve al tipo de operación anterior y se avanza en el grado de dificultad incrementando la cantidad de cifras. El control de aciertos y errores, la evaluación final de cada sesión y la adaptación a la capacidad de resolución del alumno son las tareas que controlan este tipo de programas de operaciones aritméticas.

- *Variantes de precálculo. El desarrollo de la habilidad númerica de cálculo puede iniciarse aun antes de conocer la mecánica de las operaciones básicas. Un niño de cuatro años contempla el juego que le propone la computadora. Aparecen dos rectángulos en la pantalla, uno encima del otro. En el superior se ven tres racimos de uva. En el recuadro inferior sólo aparece un racimo. El niño debe apretar una tecla prefijada tantas veces como figuras quiere que aparezcan para igualar a la muestra sugerida en la parte superior.*

Este ejercicio de precálculo puede progresar hasta lindar con la adquisición casi inadvertida de las técnicas de cálculo. Si se presentan en el monitor dos recuadros superiores, separados por el signo de la resta, con un desigual número de manzanas en cada uno, se está preguntando al alumno cuántas manzanas de diferencia hay. Se trata de una sustracción implícita. Para contestar, el niño tecleará tantas veces como número de figuras de diferencia haya entre los dibujos.
Las figuras aparecerán en el recuadro inferior y, si la solución es acertada, la computadora lo dará a conocer haciendo sonar una musiquilla alegre.

El programa es un material primario muy útil si se tiene el buen juicio —por parte de educadores y padres— de realizar una tarea de planificación didáctica, una dosificación y unos objetivos claros y cuantificables. Un camino hermoso y limpio que no conduce a ninguna parte pierde su encanto y no hay razón para recorrerlo. Esto mismo puede ocurrir con algunos programas de este tipo, que se desvirtúan o quedan en casi nada si no se desarrolla un trabajo previo y otro posterior al propio uso del programa.

Por el contrario, la ventaja de esta abundante oferta educativa reside en la posibilidad de un uso personal y descentralizado. Cualquier alumno puede repasar en el aula de tiempo libre o en el hogar aquellos temas y habilidades que no domina suficientemente. También cabe la posibilidad de que, más por interés que por obligación, se adentre en aspectos que le atraen y que aún no se le han exigido en sus estudios.

El interés general de los programas de reforzamiento de estructuras se presenta por partida doble. Por un lado, completan la actividad escolar; se introducen en las actividades corrientes del aula y se incorporan a los planes de estudio. Por otro lado, pueden ser paralelos a la actividad escolar, sin ninguna necesidad de relación, como ocurre con las lecturas o con las prácticas deportivas que los alumnos realizan por iniciativa propia en horas de ocio.

El uso de programas de refuerzo de estructuras desarrolla la función didáctica de la adquisición y/o maduración de conceptos, habilidades y técnicas, datos y ralaciones entre elementos. Esto es evidente si se examina la función de los ejemplos de programas que se han presentado. Cada uno de ellos desarrolla una especialidad de la metodología del refuerzo de estructuras; y no se agotan aquí todas las posibilidades. Las especialidades fundamentales se explican a continuación.

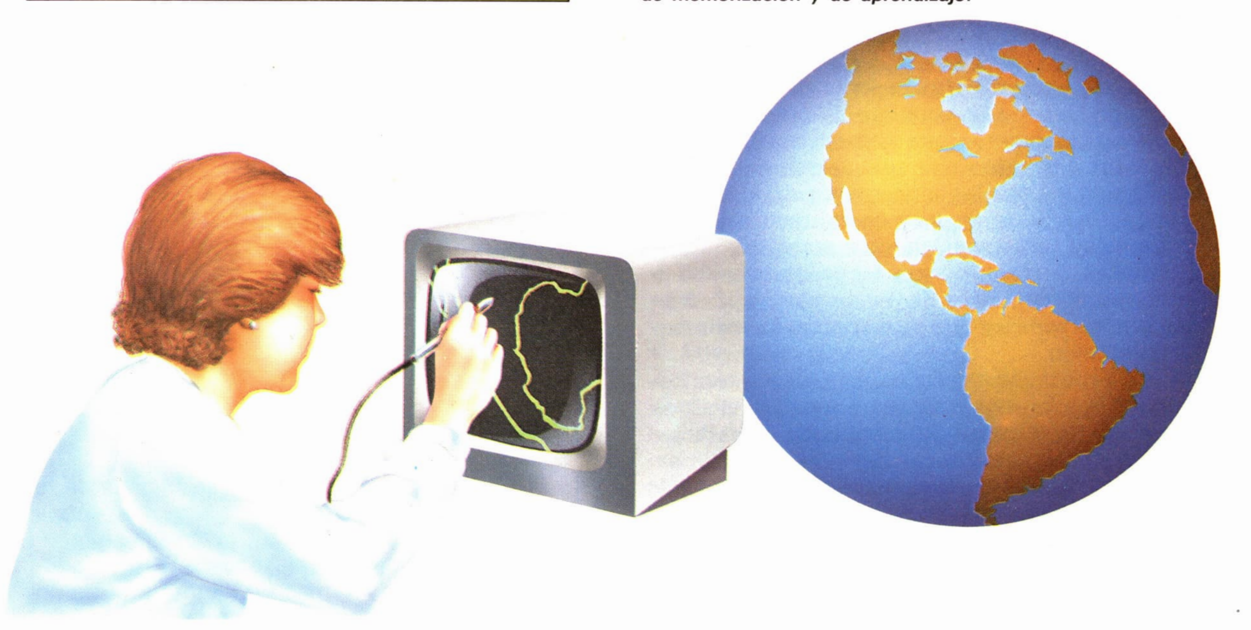
Tercer ejemplo:
Ejercicio de geografía

El programa reproduce en el monitor el mapa de un país y va presentando el nombre de las ciudades importantes. A continuación, se borran los nombres, queda el mapa mudo y se escriben de nuevo los mismos nombres. Esta repetición facilitará el reconocimiento de las ciudades. Ya sin indicación nominal, en el lugar del mapa parpadea un círculo de color y se pregunta al alumno el nombre de la ciudad señalada. El resto de la mécanica resulta similar a la de ejercicios anteriores.
Este tipo de ejercicio de reforzamiento de informaciones se puede aplicar a múltiples temas. La repetición refresca los conocimientos o, inclusive, consigue proporcionarlos por primera vez.

«Drill and practice»

El ejercicio de léxico, memorización y ortografía constituye un claro ejemplo de actividad

Abajo, ejercicio de geografía. Programas didácticos de este tipo pueden facilitar muchísimo a los alumnos las tareas de memorización y de aprendizaje.



de *drill and practice*. Esta expresión inglesa tan extendida se refiere a la forma de trabajo que hace hincapié en dos aspectos:

— la *inculcación* de unos conocimientos;
— la *práctica*, la insistencia en esos conocimientos para fijarlos.

El ejercicio en cuestión permite adquirir técnicas de memorización y practicar la ortografía de las palabras que presentan alguna dificultad. Este tipo de práctica es muy útil en el aprendizaje de idiomas para asimilar y perfeccionar pronunciación, escritura, vocabulario y fraseología.

Operaciones y técnicas

El segundo ejemplo propuesto, el de cálculo, desarrolla ejercicios para la adquisición de técnicas operatorias y aritméticas. Es una aplicación paralela a las de *drill and practice*.

Práctica de contenidos

En el ejercicio sobre geografía se despliegan técnicas para la adquisición y práctica de contenidos: datos, conceptos e informaciones generales. Este tipo de programación permite trabajar cualquier tipo de contenido de las materias de estudio en la escuela, facilitando así la superación del atraso o las lagunas que los alumnos presenten en cada caso particular.

SIMULACIÓN

La técnica de la simulación consiste en la representación gráfica, simbólica o contable de un fenómeno mediante la computadora. Ello significa que muchos fenómenos que difícilmente pueden darse en la realidad o que, de lograrse, resultarían poco aconsejables, se pueden reproducir o recrear computacionalmente sin necesidad de acudir a fuentes reales. Al decir poco aconsejables nos referimos a su carestía, a su peligrosidad, o a su inaccesibilidad o lejanía respecto del aula y los alumnos.

Con los programas de simulación no se trata ya de ofrecer ciertas informaciones y contrastarlas con el conocimiento que de ellas posee el alumno (como es el caso de los programas de reforzamiento de estructuras), sino de reproducir procesos o fenómenos complejos. El interés didáctico se centra en la comprensión y conocimiento de todo el proceso, y no sólo de algún dato o elemento final.

Los programas de simulación (si se va más allá de la superficialidad de algunos efectos visuales) requieren un trabajo más amplio y complejo que los de reforzamiento de estructuras. En ellos se encierra un micromundo, una realidad que se reproduce y que atiende a diferentes variables. Por consiguiente, no se trata de programas lineales, sino ricamente interactivos en el sentido de que las respuestas o solicitaciones del usuario pueden conducir, y de hecho conducen, como en el caso de una partida de ajedrez, a muy diferentes resultados.



Arriba, interior de un simulador de vuelo de la compañía Iberia en el aeropuerto de Madrid-Barajas. Aunque la aeronáutica aventaje muchísimo, tecnológicamente hablando, a la educación, la cabina de simulación de vuelo es un excelente banco de pruebas para su aplicación en el campo de la enseñanza por computadora.

La extraordinaria potencialidad de la simulación convierte a esta técnica computacional en una de las ayudas didácticas más prometedoras. Su uso conduce a situaciones nuevas y renovadoras. Como contrapunto, hay escasez de este tipo de programas y, aun dentro de lo que se ofrece, no pocos de los títulos son mera apariencia de simulación.

Simular el vuelo

Sin necesidad de entrar en lo que se considera estrictamente el medio escolar, el público conoce sistemas de simulación de gran calidad. Los medios de comunicación social han dado a conocer repetidamente el método de entrenamiento de vuelo para pilotos en el que interviene la simulación de vuelo. Este sofisticado programa nos ofrece un ejemplo pleno de las posibilidades metodológicas de la simulación.

La preparación de los futuros pilotos se desarrolla en tres medios distintos: el aula, la cabina de simulación y el pilotaje real.

La cabina de simulación reproduce todas las situaciones posibles durante un vuelo: todas las condiciones de visibilidad (vuelos diurnos o nocturno, niebla), maniobras (despegue, aproximación, aterrizaje, variaciones) e incidencias (fallo de motores, aterrizajes forzosos en tierra o mar, etc.). Esto significa la síntesis de todo lo que un piloto debe conocer y dominar.

Volar directamente en un avión tiene serios inconvenientes. En primer lugar, puede ser prematuro para los conocimientos del alumno. En segundo lugar, resulta muy costoso en material, carburante y tiempo. Finalmente, aun volando, difícilmente se pueden experimentar todo tipo de situaciones; tampoco se pueden realizar maniobras de despegue o aterrizaje en aeropuertos abiertos al tráfico comercial.

No sólo los aspirantes a piloto pasan por el banco de simulación. Experimentados pilotos también lo hacen cuando su compañía aérea incorpora a la flota nuevos modelos. El adiestramiento para las nuevas y complejas aeronaves también se facilita con la simulación, de manera previa a la realización de vuelos reales.

De la avanzada formación aeronáutica —que es una vanguardia tecnológica— al mundo de la educación en general y de la computación doméstica hay un trecho muy grande. No obstante, existen programas de simulación muy interesantes y no menos meritorios.

Con ánimo de establecer una línea de continuidad entre los dos ámbitos, en el ejemplo de la simulación de vuelo, cabe apuntar que ya en 1984 se comercializó de forma amplia un curioso programa para computadoras de 64K. Consistía en la simulación de un vuelo de costa a costa entre dos ciudades de Estados Unidos. Los elementos que debía controlar el «piloto» que estaba al mando del teclado no podrían compararse con los del programa profesional, pero eran lo suficientes y variados como para tener aliciente: rumbo, altitud, incidencias atmosféricas, consumo de carburante (y su dosificación según los depósitos para no desestabilizar el aparato), constatación del seguimiento de la ruta adecuada, etcétera.

La cabina de simulación de vuelo

Lo ideal es que los futuros pilotos puedan practicar en los campos de aterrizaje donde en un futuro tendrán que operar. El sistema de simulación puede sustituir el aterrizaje en aeropuertos reales, y la práctica del alumno en la cabina es tan efectiva como si se hubiera desplazado al aeropuerto de la ciudad en cuestión. Por otro lado, en la simulación intervienen todas las condiciones del pilotaje del avión. Nada se escatima, y el alumno debe realizar bien las operaciones si no quiere sufrir algún «accidente». Este último aspecto es notable. Significa que el alumno tiene la responsabilidad del control del aparato y aprende de sus aciertos y sus errores. Si el aspirante estuviera en vuelo real, el método de instrucción no permitiría ningún error. La preparación intensa y extensa para que esto no ocurra se realiza con verdadera eficacia en la cabina de simulación de vuelo.

Cuidar un invernadero

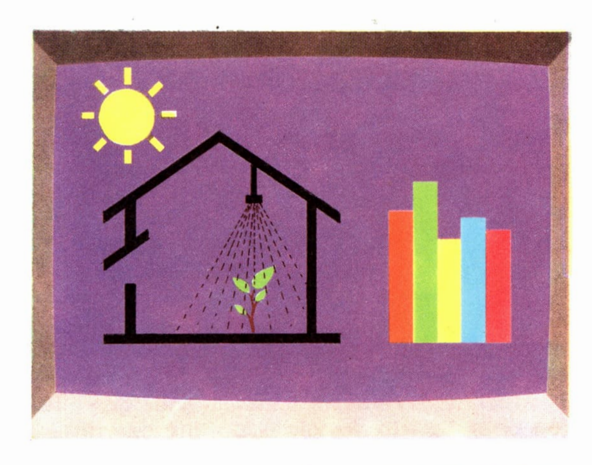
No es fácil —ni conveniente— establecer una división entre los programas según sean educativos o no. En muchos casos, el aprovechamiento escolar del software no depende tanto del programa en sí como de su enfoque. Lo que sí resulta evidente para todos es que la labor educativa no acaba ni comienza en el uso de horas de computadora o de unos programas aprovechables. Es preciso trabajar con una cierta continuidad y en relación con actividades que signifiquen algo más que el uso del teclado, como escribir, pintar, expresarse en público, indagar en la realidad, etc. De ahí que la forma de trabajar que integra tecnología y prácticas tradicionalmente creativas resulta inmejorable.

Dicho esto, se puede pasar al conocimiento de ciertos elementos de simulación útiles para nuestro propósito. Un ejemplo real se encuentra en la construcción de un invernadero; un invernadero simulado, por supuesto. Construirlo realmente exigiría mucho tiempo, materiales, espacio libre, con sol y agua, y la paciencia para esperar el resultado de las experiencias. En cambio, con la simulación del invernadero es posible experimentar con diversas especies, extraer conclusiones rápidamente y hacer que todos los alumnos participen en sus pruebas en cada caso particular.

El invernadero se presenta en la pantalla de manera esquemática. Una semilla está plantada y su germinación y el desarrollo de la planta dependerán de unas cuantas variables que regula el alumno, a saber:

- humedad;
- aireación;
- luz.

Las combinaciones de estas variables determinan distintos resultados, que se pueden evaluar. Uno de los objetivos puede consistir en alcanzar el máximo desarrollo (en altura) de la planta en el mínimo tiempo posible. Otro puede ser comprobar cuánto tiempo sobrevive en ausencia de algunos de los elementos. Para la humedad, una tecla controla el grifo de agua; otra tecla abre o cierra de forma gradual un ventanal; la iluminación depende del ciclo natu-



Esquema de un invernadero: El programa permite al alumno controlar las condiciones ambientales en función del objetivo que se persigue.

ral del día y puede mitigarse. El grado de humedad se da en dígitos en pantalla; una regla indica la altura de la planta; un contador marca los días de crecimiento de la planta.

Realizar una expedición a través de todas las asignaturas

Si el programa del invernadero se circunscribe al área de las ciencias naturales (como otros se ocupan de física, geometría, dibujo, gramática, etc.), existen otros programas —realmente escasos— que permiten un juego educativo mucho más amplio, profundo y largo a la vez. Han sido concebidos para los últimos cursos de enseñanza primaria. Su plan es tan amplio y flexible como fascinante. Se trata de organizar un trabajo, mediante un programa de simulación, en el que intervienen la mayoría de las asignaturas del curso. Aquí la computación no se utiliza para estudiar computación ni para una simple asignatura, sino para organizar un saber que relaciona todo lo que, de otro modo, el alumno conoce por separado y de forma inconexa.

La idea de este tipo de programas de simulación es tan nueva y tan poderosamente educativa que no es muy común su aplicación. Ello se explica porque exige una programación del curriculum distinto al usual.

Un ejemplo de los programas que persiguen este propósito es el de la *Expedición a Saqqara*, escrito por profesores ingleses de enseñanza primaria. Consiste en la simulación de una expedición al mítico valle de Saqqara, en el Antiguo Egipto; se trata de una expedición arqueológica. Su objeto es buscar los restos de tumbas y materiales de la sociedad faraónica. La tarea se desarrolla en varias clases semanales.

El programa propone a los alumnos esta aventura, expone los objetivos y sugiere las tareas complementarias. En primer lugar, aquéllos deben documentarse acerca de lo que van a buscar, el período histórico que van a rastrear y la geografía y formas de vida del Egipto actual. Estas informaciones se extraen de libros de consulta. De modo paralelo, los alumnos elaboran unos presupuestos y redactan escritos de solicitud de ayuda que, supuestamente, van dirigidos a los organismos adecuados para recibir subvenciones. Con ello se ejercita la aritmética, la contabilidad y la expresión gramatical; al mismo tiempo, se realizan todos los pasos que se seguirían para una expedición real.

Los profesores asumen el papel de las instituciones a las que teóricamente van dirigidas las peticiones de ayuda y dan luz verde a la iniciativa. Nominalmente, pues, los alumnos ya disponen de un presupuesto y pasan a otros preparativos. La computadora va memorizando todo lo que se realiza, computa el presupuesto de que se dispone y va anotando los gastos.

ÁMBITO Y OBJETIVOS

La simulación abarca todos los aspectos organizativos de la expedición (tipo de transportes, costo, duración del viaje) y de la realización de las excavaciones. Para esto último el programa ofrece en pantalla unos cuadrantes en los que los alumnos, por grupos, comenzarán a «excavar». En ellos irán encontrando restos arqueológicos que deben estudiar, dibujar, cotejar con las reproducciones fotográficas de libros de arte, etc. Mientras tanto, los profesores introducen algunos imponderables que deben ser resueltos: un expedicionario enfermo (hay que trasladarlo a la ciudad más cercana y desplazar algunos compañeros), un camión ha sufrido una avería, etcétera.

La descripción del programa de simulación de la «Expedición a Saqqara» requiere bastante más que unas simples líneas, máxime si su duración es similar a la de una asignatura de curso. No intentaremos describirlo. Lo que destaca es la originalidad de la idea; con la intervención de la computadora y un complejo programa, se logra exactamente lo que se trata de conseguir en la escuela: la simplicidad de un quehacer que anima a moverse con seguridad y provecho en los campos de estudio.

En el caso de la «Expedición a Saqqara», es evidente que se logra trabajar de forma conjunta e interdisciplinada con la mayoría de las materias del curso: *matemáticas, geografía, historia, arte, tecnología, literatura* y *dibujo*. A esto se suma la nada despreciable experiencia que significa enfrentarse a problemas de la vida práctica como la redacción de instancias, planificación de detalles materiales, etc. Y, lo que es más meritorio, los contenidos de las materias y las actividades complementarias se practican sin advertirlo los alumnos, de forma natural.

La simulación aporta una nueva forma de conocer

Los programas valiosos de simulación evidencian unas posibilidades didácticas que resultan nuevas entre lo nuevo. Si la aplicación de la tecnología computacional puede abrir un camino distinto para aprender y comprender, la simulación se halla en la parte más alta de este movimiento de renovación.

La simulacion permite realizar una síntesis de la realidad o el fenómeno estudiado. Ello clarifica, vierte luz sobre las cosas que se extienden en el tiempo y el espacio y que, por ello, escapan a una visualización completa por el alumno. La simulación puede acelerar ciertos procesos y acercarlos al aula, lo cual es bastante práctico si se considera que la posibilidad de conocer las cosas sobre el terreno está muy restringida con los horarios y los planes escolares.

Todo lo anterior tiene su valor, pero no es lo concluyente. Aquello que proporciona mayor relevancia a la simulación es el hecho de que se admite lo hipotético; permite trabajar en hipótesis, en un laboratorio conceptual que experimenta acerca de algunas situaciones que, de otro modo, no podrían conocerse. El alumno puede adentrarse en la comprobación de una posibilidad y conocer qué podría ocurrir.

Por último, es preciso añadir una consideración importante: no excluir la realidad. El hecho de disponer de programas de simulación y medios electrónicos no debe nunca conducir a olvidarse de la experimentación directa sobre las cosas. De otro modo, el resultado sería absurdo. La simulación es un instrumento útil si no nos aparta de aquello que es accesible y también más cercano.

OTRAS TÉCNICAS COMPUTACIONALES

Estructuración de problemas

Parece claro que, dado que los objetos y las condiciones de la enseñanza van a cambiar mucho en pocos años, lo más importante no es formar a los alumnos en unas tareas muy concretas, sino desarrollar su capacidad para llevar a cabo tareas diferentes y aun otras que hoy no se conocen. Tal es la razón de que resulte prioritario aprender procedimientos o programas para resolver problemas antes que resolver los propios problemas. Indudablemente, la tendencia se ha invertido. No interesa tanto buscar mecánicamente la solución de problemas como descubrir formas de resolución. El tipo de técnica que hace hincapié en este planteamiento es el de la estructuración de problemas.

¿En qué consiste la estructuración de problemas? Supongamos que nos hallamos en el aula operando con lenguaje LOGO, que resulta dúctil para los niveles de enseñanza primaria y secundaria. Se solicita a los alumnos que dibujen un cuadrado mediante la «tortuga» gráfica de LOGO. A continuación se les pide que diseñen un programa que sirva para construir cualquier tipo de cuadrado, sin estar limitado a una medida dada. Cuando se logra esto, se está aplicando la variable matemática; no es tan importante verbalizar el concepto de variable como interiorizar su estructura.

De manera similar, pero ampliada, la técnica de estructuración de problemas puede continuar indicando a estos alumnos que dibujen una casa, también con instrucciones LOGO. Lo normal es que se ejecute el dibujo de forma global. Para ello, puede sugerirse que se enfoque la tarea de manera modular, esto es, que se trabaje por partes y figuras: la fachada, un cuadrado; el tejado, un triángulo; las ventanas, cuadrados; la puerta y la chimenea, rectángulos. Tras varias tentativas con mayor o menor éxito, se adquiere la técnica de la resolución modular de los problemas y de construcción de un *procedimiento*.

De esta manera, el lenguaje computacional (como ocurre con LOGO) se ofrece a los alumnos como un juego de piezas o mecano que permite grandes resultados con la combinación comprensiva y expansiva de pequeños elementos.

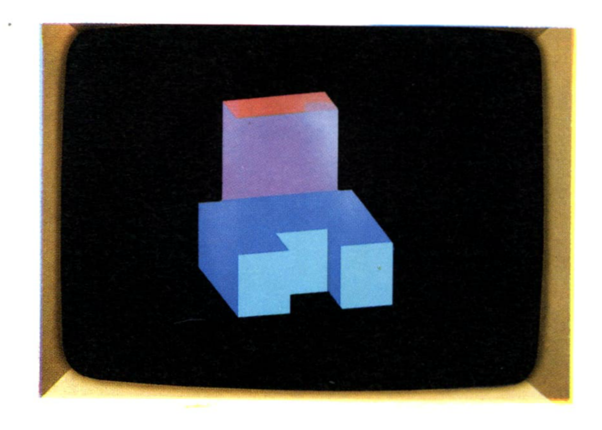
Manejo de información

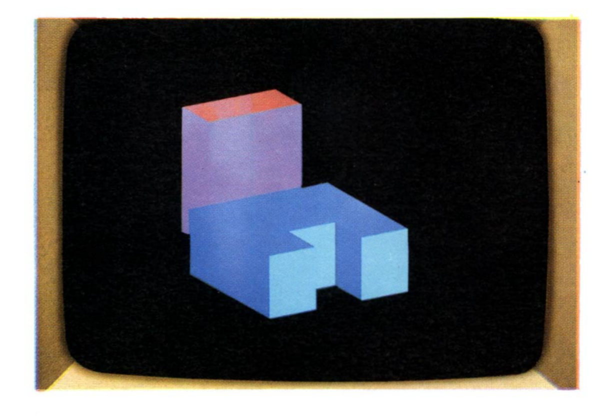
La disponibilidad de abundantes fuentes de información en grandes computadoras abre la puerta a otras formas de documentación distintas a las que se encuentran en los libros y en las bibliotecas. El manejo de información que se pone a disposición de los usuarios de terminales de consulta tiene un mayor aprovechamiento que el del puro aumento cuantitativo.

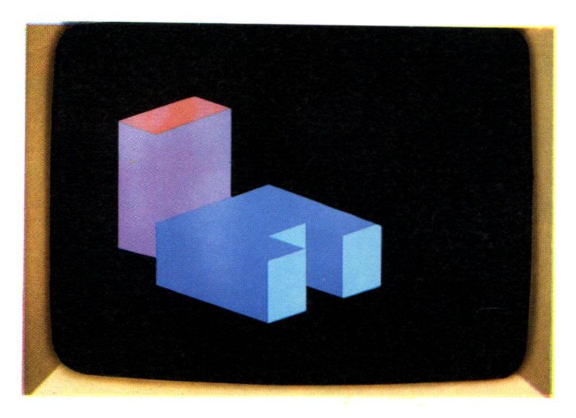
Es evidente que la abundancia de datos aporta mayor comodidad y favorece el trabajo de documentación, pero no lo asegura. Para que sean relevantes, los cambios cuantitativos deben ir acompañados de cambios cualitativos. Lo cualitativo es equiparable a las tácticas de tratamiento de información en el medio escolar. Sean cuales sean en concreto, lo común a to-

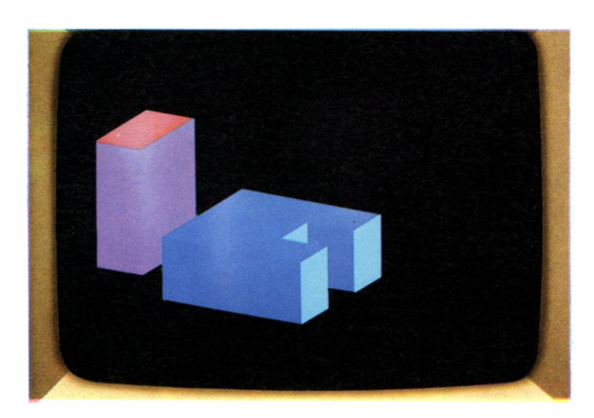
Simulación y experimentación

La simulación aporta una nueva forma de conocer. Simular no es otra cosa que obtener información de experimentos inventados. Es una forma de conocer diferente a la simple experimentación, que consiste en probar observaciones próximas, materialmente inmediatas.









Los ingenieros y arquitectos aprovechan plenamente en su actividad profesional la posibilidad que ofrecen las computadoras de obtener distintas perspectivas de un cuerpo o de desarrollar una perspectiva a partir de un dibujo en planta y alzado. Pero, los escolares también pueden beneficiarse enormemente de esta capacidad para gráficos de que disponen hoy en día muchas computadoras, incluso de pequeño tamaño, para desarrollar su sentido geométrico. Existen así mismo programas que permiten la resolución gráfica de problemas geométricos y que, a través de un proceso de control de aciertos y errores, permiten a los alumnos trabajar «directamente» en la resolución de este tipo de problemas.

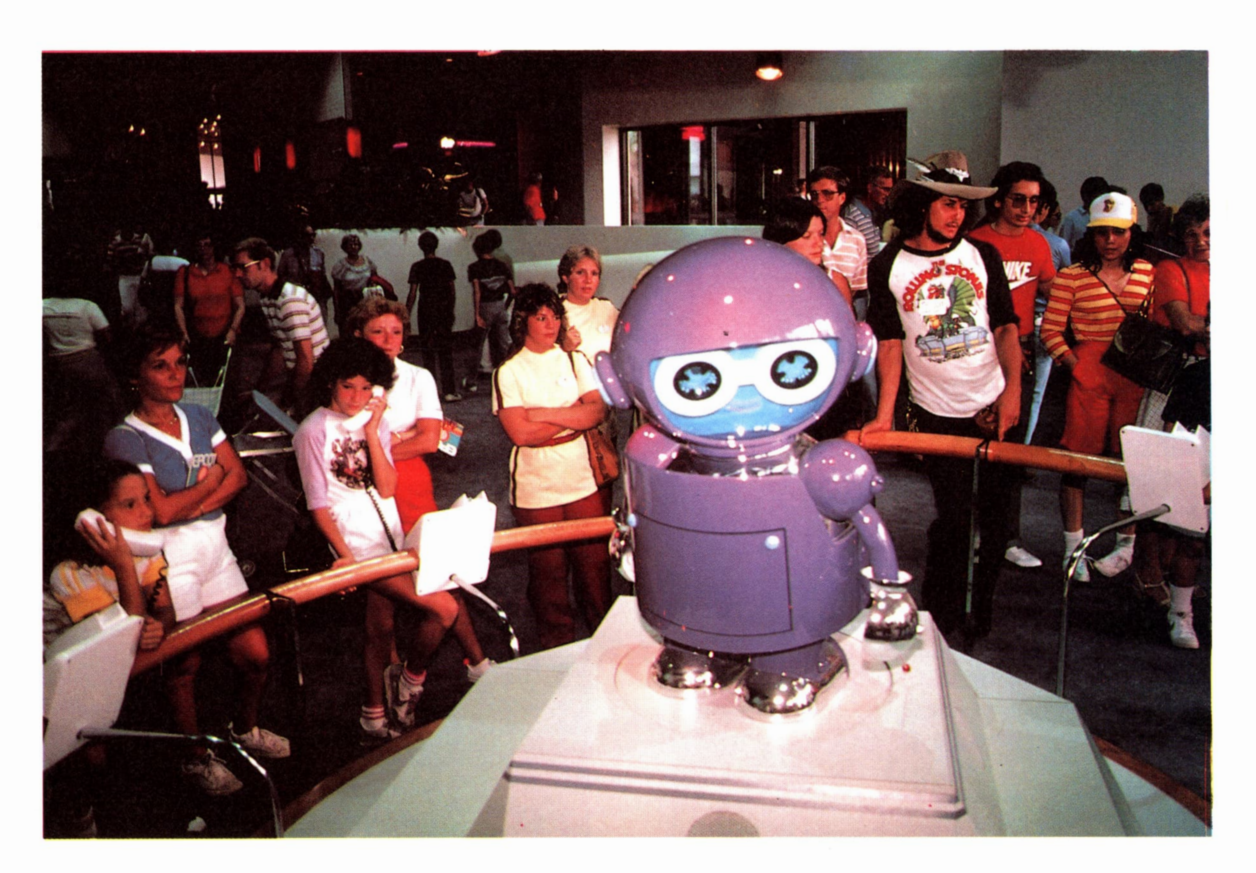
das estas técnicas estriba en el desarrollo de capacidades de búsqueda, selección, cotejo y producción final de información elaborada.

Dentro de esta gama, es una técnica usual la confección de cuadros estadísticos de los elementos de un cuerpo informativo. Este tratamiento contable no sólo interesa para realizar ejercicios matemáticos, sino para conocer temas de cualquier asignatura. La estadística permite cuantificar y establecer graduaciones entre diversos fenómenos, como por ejemplo los movimientos de población en un lugar, las actividades económicas a lo largo de un período, los tipos de trabajos y profesiones, los indicadores de condiciones de vida, etc. Esta enumeración correspondería a un tema de geografía económica y de historia.

Control tecnológico y robótica

El uso de sistemas computacionales para el control de máquinas es también digno de atención en la escuela; se le denomina control tecnológico. Las aplicaciones educativas presentan dos variantes fundamentales:

- la máquina comandada mediante el teclado de la computadora;
- la máquina con sistema de control autónomo.



Los robots, como el de la compañía Bell que aparece en la fotografía, atraen siempre la atención del público infantil y juvenil en las ferias y exposiciones.

Un artilugio mecánico, conectado a la computadora como periférico normal, ejecuta los movimientos que se instruyen. En la pantalla, el cursor se desplaza según el programa. Sobre un plano, de forma idéntica, se desplaza el aparato merced a las ruedas de que está dotado. Su aspecto puede ser el de un carro o el de una tortuga; la tortuga está más generalizada. Disponen estos aparatos de un bolígrafo o pluma que dibuja sobre el plano de desplazamiento; así queda trazado su recorrido. Estos aparatos son sencillos robots que disponen de varios motores para realizar las maniobras más elementales: avanzar, retroceder, girar, bajar-subir la pluma.

La otra variedad está representada por las máquinas que llevan incorporado el mecanismo de control. Se presentan como camioncitos o tanques y se comercializan como juguetes. Su uso didáctico estriba en la misma actividad que en la anterior variedad: la programación de sus movimientos de acuerdo con un plan previsto. El sistema de control autónomo permite disponer una secuencia programable. La adecuación entre los movimientos previstos y su cumplimiento con precisión favorece la interiorización de los conceptos y técnicas de programación desde edades muy tempranas, y en un ambiente de juego sumamente estimulante.

PROGRAMAS ACTUALES Y PROGRAMAS PARA EL FUTURO

La evaluación de los programas que se ofrecen no es fácil. Para conocer su calidad es preciso someterlos a prueba y no basta la superficial comprobación de sus elementos generales. En estos casos de apresuramiento, la edición en colores llamativos y sonidos efectistas juega un papel contraproducente, ya que una atractiva presentación puede llamar a engaño, al ir en detrimento de un pobre contenido o de una insuficiente planificación.

No existen las recetas certeras para calibrar la calidad de un programa o de un paquete de software educativo. A lo sumo, se conocen enumeraciones de aspectos que intervienen en la elaboración del juicio de calidad. Nada sustituye a la experiencia.

Aun así, algunos puntos de referencia fundamentales pueden servir de orientación. El primero es el que atañe a la documentación complementaria para uso del profesor. Lamentablemente, es muy general el olvido de esta condición inexcusable. La documentación debe indicar sugerencias del método de enseñanza, objetivos que se persiguen, formas de integración

en el curriculum y desarrollo de las cuestiones que se tratan en el software.

Junto a la documentación o guía didáctica, deben cumplirse otros requisitos, como son:

- la indicación de edades y el nivel educativo a que está destinado;
- la edición clara y expresiva en pantalla;
- la ramificación según dificultades y la inclusión de opciones de ayuda;
- la educación del método al contenido;
- el efecto (potenciador) que ejerce el planteamiento y la presentación en la comprensión y maduración del tema;
- la adaptación del lenguaje de las instrucciones a los usuarios;
- la utilización de ejemplos y demostraciones;
- la posibilidad de que el alumno controle el ritmo del programa, cómo interrumpirlo y continuarlo en otro momento;
- la educación del diálogo entre el programa y el usuario y las características de la retroalimentación o feedback;
- la elaboración automática de un archivo de datos sobre el trabajo del usuario;
- la posibilidad de que el programa no sea hermético y el alumno ejerza su creatividad.

La relación de las condiciones que ha de reunir la producción de software no pueden agotarse aquí. El cierre de la anterior enumeración con la indicación de la creatividad personal es intencionado. Es un aspecto generalmente desatendido. Se suele entender —erróneamente— que un programa cerrado, invariable en su estructuración, es la mejor manera de conseguir un rendimiento educativo. Se olvida que guiar firmemente al alumno no siempre es conveniente y que, en ocasiones, es necesario ofrecer opciones reales de creatividad, de apertura a formas personales y particulares de operar y de resolver.

El software que actualmente existe necesariamente tiene que evolucionar; de forma que cuesta imaginar, al mismo ritmo que lo haga el hardware, y que se hagan realidad los presupuestos técnicos de la quinta generación. Por tal razón, lo que puede esperarse de las aplicaciones computacionales a la educación es esperanzador. La escasa oferta de calidad y la precaria planificación didáctica actual pueden ser superadas con la instrumentalización de programas más ricos, potentes, profesionales e interactivos. Así, la utilización de mejor software vendrá acompañada del uso en la escuela de la computación y la telemática.

El dibujo ilustra una casi segura alegoría del futuro: La educación avanzará cada vez más por los caminos abiertos por la computación.



La inteligencia artificial y la quinta generación

Con un gran sentido del humor se suele bromear acerca del futuro de la tecnología con la siguiente frase: «Predecir es difícil, pero mucho más difícil es predecir el futuro.» Esta expresión desenfadada revela una actitud severa y seria. Se puede intentar prever cuáles van a ser los efectos de las invenciones que están a punto de llegar al mercado y qué novedades tecnológicas configurarán la sociedad del futuro. Ello no sólo es lícito, sino, además, muy interesante. Aun así, lo cierto es que ni siquiera los mejores especialistas en las diversas tecnologías pueden ofrecer a ciencia cierta una visión medianamente aproximada de lo que nos deparará el futuro; de ahí que ironicen con la predicción y su imposibilidad.

Evidentemente, los especialistas están trabajando hoy en lo que el público conocerá a corto y medio plazo, y todo este conjunto de ambiciosos proyectos no puede cogerles por sorpresa. Se afanan en su diseño desde hace tiempo; pero hay un salto desconocido entre el perfil de la tecnología venidera y la forma como ésta puede influir y determinar la organización social, el trabajo y el conocimiento. Por consiguiente, hay un salto entre dos elementos:

— la tecnología (los aparatos, computadoras y robots, entre otros);
— la implantación social.

Del segundo elemento sólo cabe aventurar conjeturas muy imprecisas. Y del primer elemento tan sólo parcialmente se pueden elaborar decisiones aproximadas, pues el resto está sometido a factores aún por determinar.

PROTOTIPOS HOY, REALIDAD COTIDIANA MAÑANA

Se dispone ya de muchos y variados ejemplos del desarrollo tecnológico que está en ciernes. En ellos juega un papel destacado la computación, la microelectrónica, los microprocesadores, la robótica y los sistemas expertos, entre otros aspectos. Estos ejemplos nos presentan con cierta aproximación una visión futurista que comienza a hacerse cotidiana o a experimentar un acercamiento rápido a nuestro presente. Veamos algunos de estos aparatos que o bien ya pueden encontrarse en el mercado, o bien han sido previstos y considerados como un objetivo asequible.

Sistemas domésticos de control

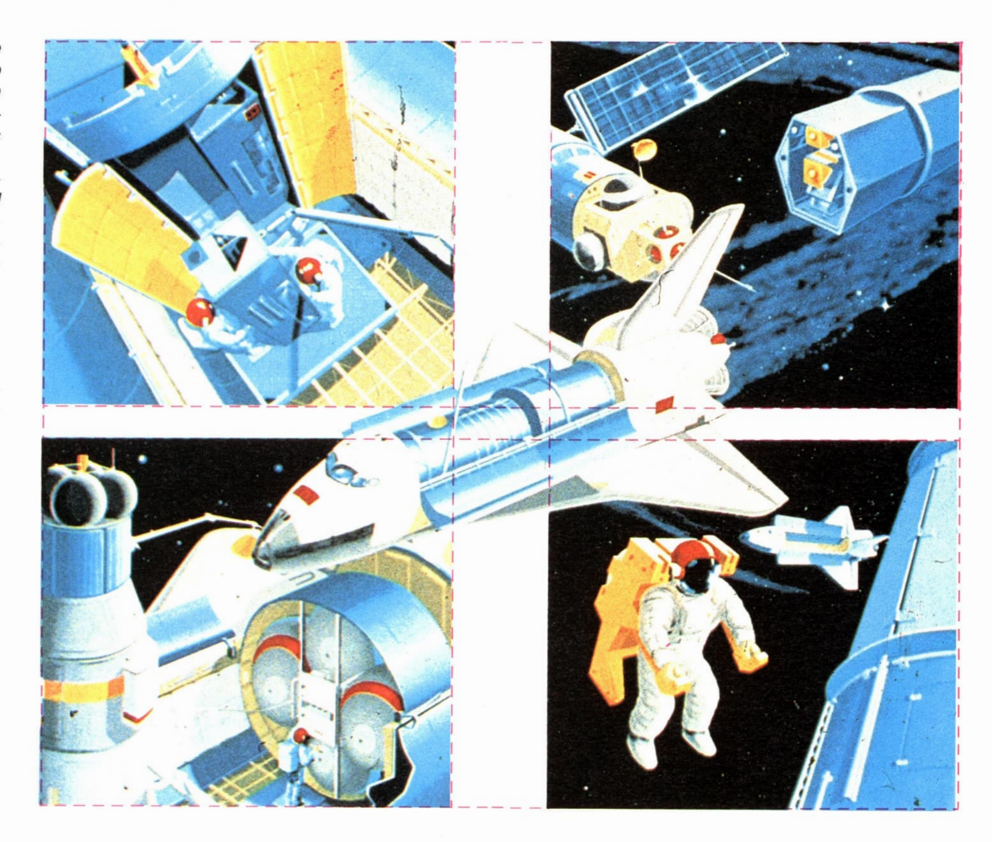
Se conocen ya los primeros modelos de sistemas domésticos de control. Consisten en mecanismos de control remoto diseñados para su uso en domicilios particulares. Con un sistema de este tipo y una instalación adecuada de periféricos, es posible controlar y operar sobre todos o casi todos los elementos de la casa. El sistema dispone de una unidad central que permite programar digital o gráficamente las funciones deseadas y ordenar su ejecución de manera inmediata o diferidamente. Una instalación o red especial transmite las señales codificadas que emite el módulo central. Las funciones dependen de la capacidad del sistema, sin otro tipo de limitación. Desde este módulo o unidad central se pueden conectar los electrodomésticos (todos o algunos determinados), encender o apagar las luces, abrir o cerrar ventanas y puertas, descongelar el frigorífico, poner en marcha la cafetera, regular la calefacción o el aire acondicionado, supervisar el funcionamiento de la lavadora o del horno de cocina, preparar el baño a la temperatura y hora prefijadas.

Los sistemas domésticos de control también pueden realizar tareas no requeridas de modo específico, como detectar fugas de gas y prevenir al usuario y/o cerrar la llave de paso, avisar a los bomberos en caso de fuego —mediante la instrumentalización de un detector de humo—, regir un sistema de seguridad, etc. Así mismo, además de la programación directa del usuario, el sistema de control también funciona de forma remota mediante el canal telefónico. El propietario puede activar por teléfono, sea cual sea la distancia a que se encuentre, su sistema doméstico de control o algunas de sus funciones. Un código secreto le asegura el acceso exclusivo. Como complemento de lo que hasta ahora ofrecen, estos sistemas pueden informar telefónicamente al propietario ausente de las incidencias que se produzcan.

Automóviles

Una automatización «inteligente» de este tipo se está introduciendo desde la década de los ochenta en los automóviles. No sólo se trata de las computadoras de a bordo, que controlan partes fundamentales del vehículo y que informan verbalmente de las incidencias, sino de

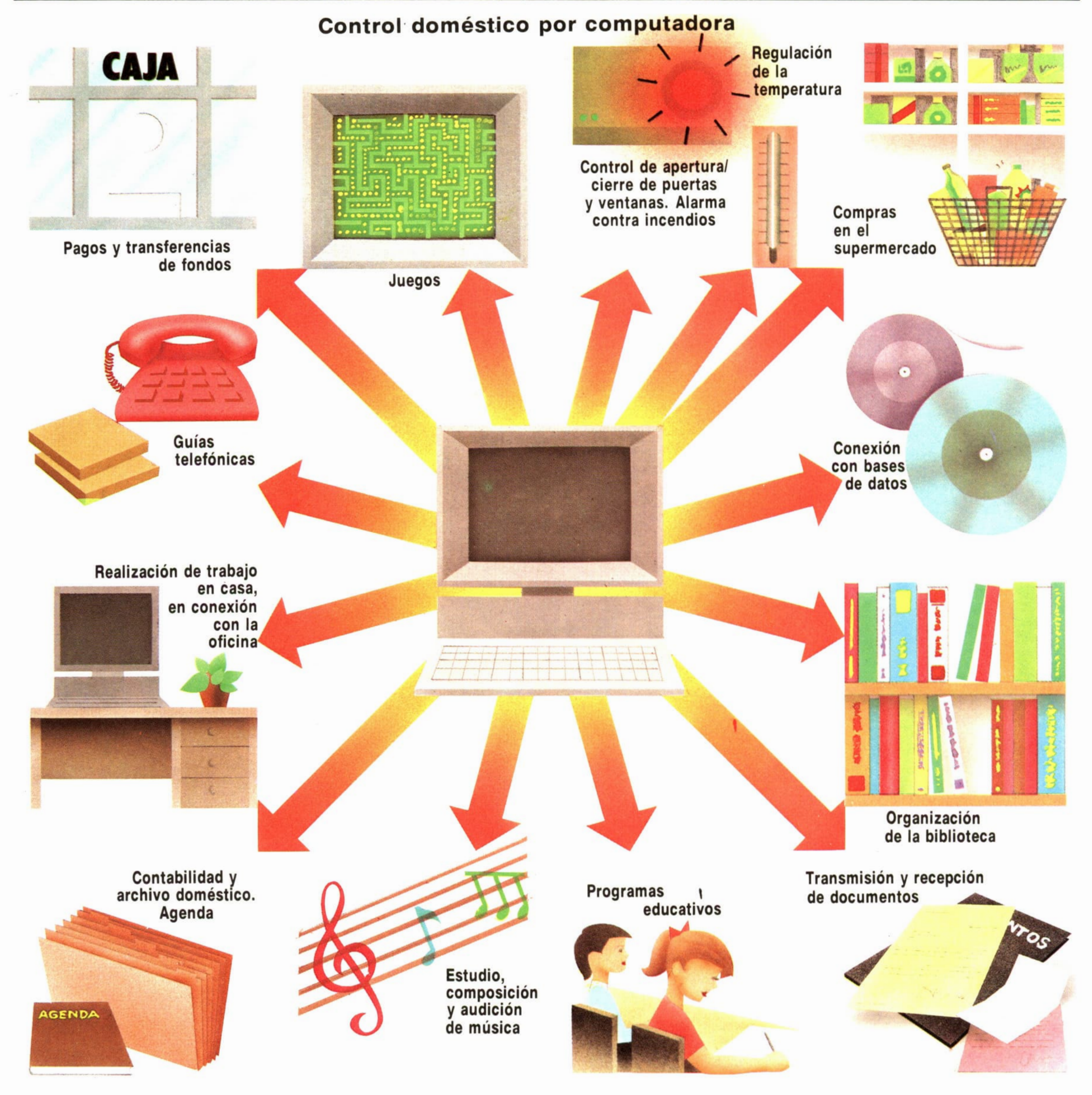
La historia del hombre puede ser enfocada como la de un largo camino, a través del progreso tecnológico, del campo de las posibilidades humanas. Muchos de los sueños ancestrales de la humanidad —el de volar, por ejemplo— se han visto cumplidos gracias a un invento —el del avión, en nuestro ejemplo. Así, puede comprenderse la revolución que supone el desarrollo de la microelectrónica, ya que ofrece la posibilidad de ampliar la capacidad humana en el orden motriz —los vuelos espaciales dependen de esta tecnología—, en el orden intelectual —inmensa capacidad de las computadoras para el cálculo y el procesamiento de datos— y, quizás, sensorial —no es difícil imaginar la implantación de microprocesadores en el organismo humano, como muestra, con suma elocuencia, el bebé electrónico de esta página.



El asado se hará solo y con una simple llamada telefónica

El primer modelo de sistema doméstico de control, aún rudimentario, comenzó a comercializarse en Estados Unidos en 1979. Cuando comience el siglo XXI, es muy posible que las funciones incorporadas sean mucho más amplias, centralicen todos los procesos computacionales (no sólo los de control) y se programen en lenguaje natural. Esto último significa que la expresión en voz alta de una orden es suficiente para que el sistema la reciba y la ejecute, y ya no sea necesario apretar botones o teclear letras y números. Incluso se piensa en una descentralización de las funciones y que el usuario pueda dirigirse a cualquier aparato y ordenarle qué tiene que hacer. Así como algunos electrodomésticos, como por ejemplo las lavadoras, están automatizadas, así lo estarán otros y, con la incorporación de microprocesadores programables, poseerán capacidad para «discernir» tareas e intensidades. Ello introduce una cualificación sustancial. Dicho a modo de ejemplo, cabe imaginar un horno de cocina que impida que por descuido se queme o se pase el asado que en él se está haciendo.

Control doméstico por computadora



aquellas aplicaciones especiales que afectan a la seguridad mediante automatismos muy eficaces. Es el caso, entre otros, del control de velocidad del vehículo al tomar una curva y la regulación automática de la velocidad para que no peligre la estabilidad ni el dominio del automóvil. Esta aplicación microelectrónica se convierte en el «ángel de la guarda» que evita vuelcos o salidas de la carretera por exceso de velocidad en virajes.

El automóvil del futuro no sólo tendrá automatizados sus elementos mecánicos y funciones internas, sino que los microprocesadores controlarán elementos externos a él, los elementos del medio. Para ello dispondrá de radar y de un sistema computacional que advertirá al conductor de la posibilidad o imposibilidad de realizar un adelantamiento. También este sistema estará alerta para frenar automáticamente en situaciones de peligro inminente de choque. El ámbito del transporte resulta tan atractivo que mueve a imaginativos proyectos. Y resultaría fácil adelantar proyectos que seguramente se alcanzarán cuando la ciencia ficción se convierta simplemente en ciencia.

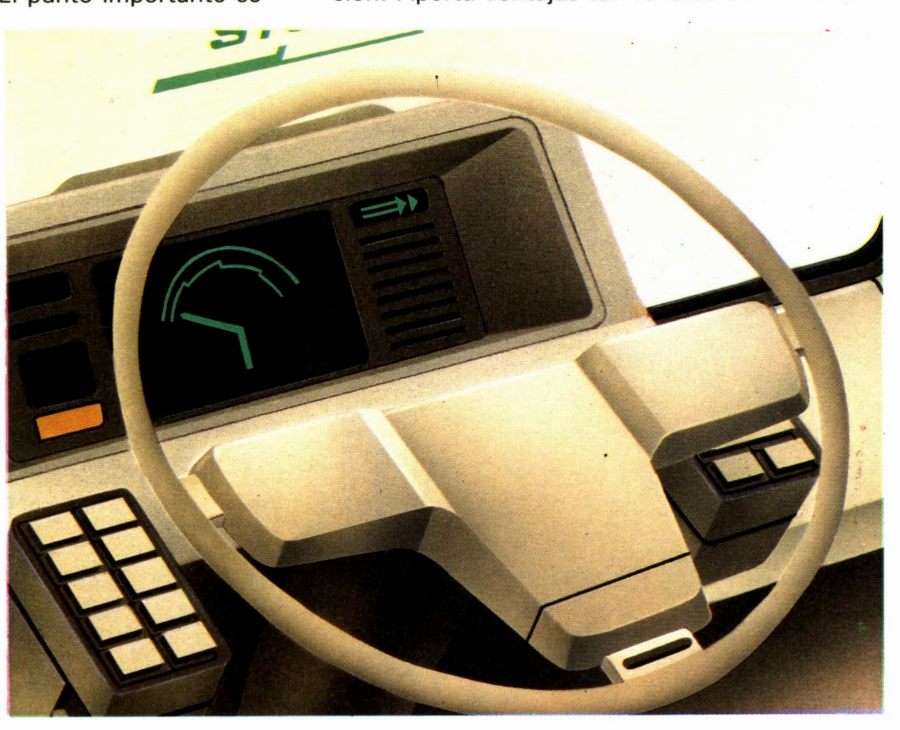
Robots

A la década de los ochenta pertenecen progresos en robótica verdaderamente notables. Una tarea tan simple como la de quitar el polvo con una aspiradora y esquivar convenientemente los obstáculos que haya, no se programa tan fácilmente en un robot. El punto importante es la detección de los obstáculos (que no siempre son los mismos ni están en el mismo sitio) y la maniobra para eludirlos y seguir trabajando con la aspiradora. En comparación, los robots industriales, que realizan operaciones muy precisas y a veces complejas, no plantean tanta dificultad en su diseño y fabricación. La razón de ello estriba en la fijeza de sus respectivas tareas. ¡Limpiar el polvo del suelo de un salón es más difícil que ajustar piezas en una cadena de montaje de automóviles!

La experimentación en operaciones quirúrgicas con robots abre nuevos campos tan positivos como esperanzadores. La cirugía requiere de los médicos una habilidad, precisión y decisión muy cualificadas. La asistencia de ingenios puede complementar algunas de las condiciones que el trabajo exige.

En operaciones delicadísimas, como las de cerebro, el robot puede aportar mayor fiabilidad. Últimamente, se ha logrado utilizar estas máquinas para realizar el cálculo de los ángulos de incisión de los instrumentos de corte y reconocimiento en operaciones cerebrales; así mismo, su operatividad se extiende a la dirección y el manejo del trepanador quirúrgico para penetrar el cráneo y de la aguja de biopsia para tomar muestras del cerebro. Estos instrumentos se utilizan para obtener muestras de tejidos de lo que se suponen tumores que presentan un difícil acceso, para lo que resulta esencial la intervención del robot.

El progreso de estas aplicaciones va más allá de la mejora de las condiciones de intervención. Aporta ventajas tan revolucionarias como

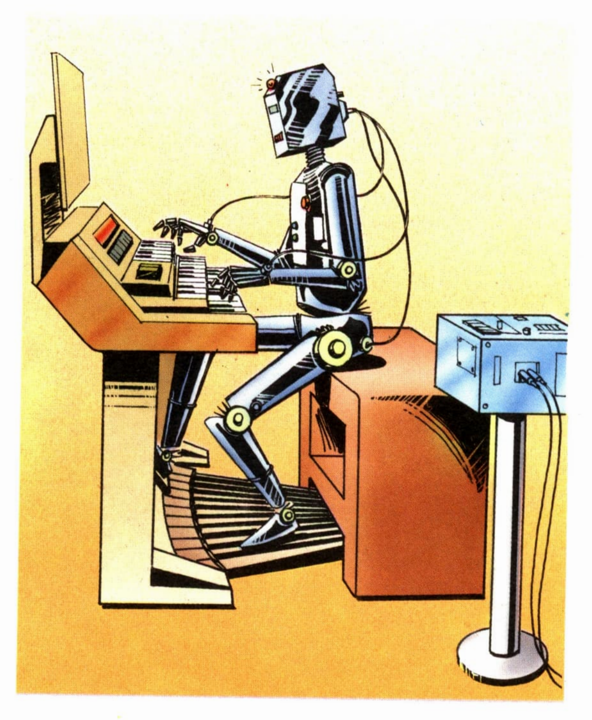


En el dibujo de la izquierda se esquematizan algunas de las funciones que puede cumplir una computadora doméstica. No parece lejano el día en que la computadora doméstica se convierta en el centro a través del que pasen todas las comunicaciones del hogar con el exterior y en el órgano regulador tanto de las condiciones ambientales de la vivienda como de todos los electrodomésticos.
A la derecha: salpicadero de automóvil computerizado. En muchos coches actuales el conductor puede informarse del funcionamiento de su vehículo sin necesidad de realizar una comprobación directa, y muchas partes fundamentales del motor incorporan microprocesadores que regulan su funcionamiento.

El robot pianista y el robot retratista

Los prototipos de robots que se presentaron en la Feria Internacional de Tsukuba (Japón), en 1985, causaron asombro en los visitantes. Un robot antropomórfico (con cierto parecido físico con un humano) era capaz de ejecutar al órgano piezas de autores clásicos. Se le colocaba la partitura ante el teclado y la leía con una cámara de televisión que llevaba en su «cabeza». Con las manos y los pies accionaba teclados y pedales de forma muy satisfactoria. Otro robot de aspecto más informal desplegaba sus habilidades de retratista. Los visitantes que posaban como modelo eran dibujados por la máquina. Una cámara de televisión captaba la figura y los rasgos de cada persona. Estos datos eran analizados computacionalmente y codificados, de manera que el brazo trazador dibujaba sobre el papel al modelo con notable parecido. Y aún otro tipo de aplicación robótica: el teatro. Unos pequeños robots representaban obritas ante el público con la ayuda de su idéntico y simbólico moldeado, graciosos movimientos, luminotecnia y bandas sonoras sincronizadas.

Prototipo de robot pianista presentado en la Feria Internacional de Tsukuba (Japón), en 1985.



el hecho de prescindir de anestesia general y el drástico acortamiento de la convalecencia.

Estas ayudas son dirigidas y controladas por el médico; no actúan autónomamente, y su futuro inmediato resulta amplio. La robótica puede servir para guiar rayos láser, reparar vasos sanguíneos, aplicar sustancias radiactivas en tumores o asistir en la intervención de ligamentos y cartílagos lesionados.

¿PUEDE PENSAR UNA MÁQUINA?

Esta pregunta tan simple plantea unos problemas tan grandes que, posiblemente, nunca se llegue a un acuerdo completo entre las distintas respuestas que se proponen.

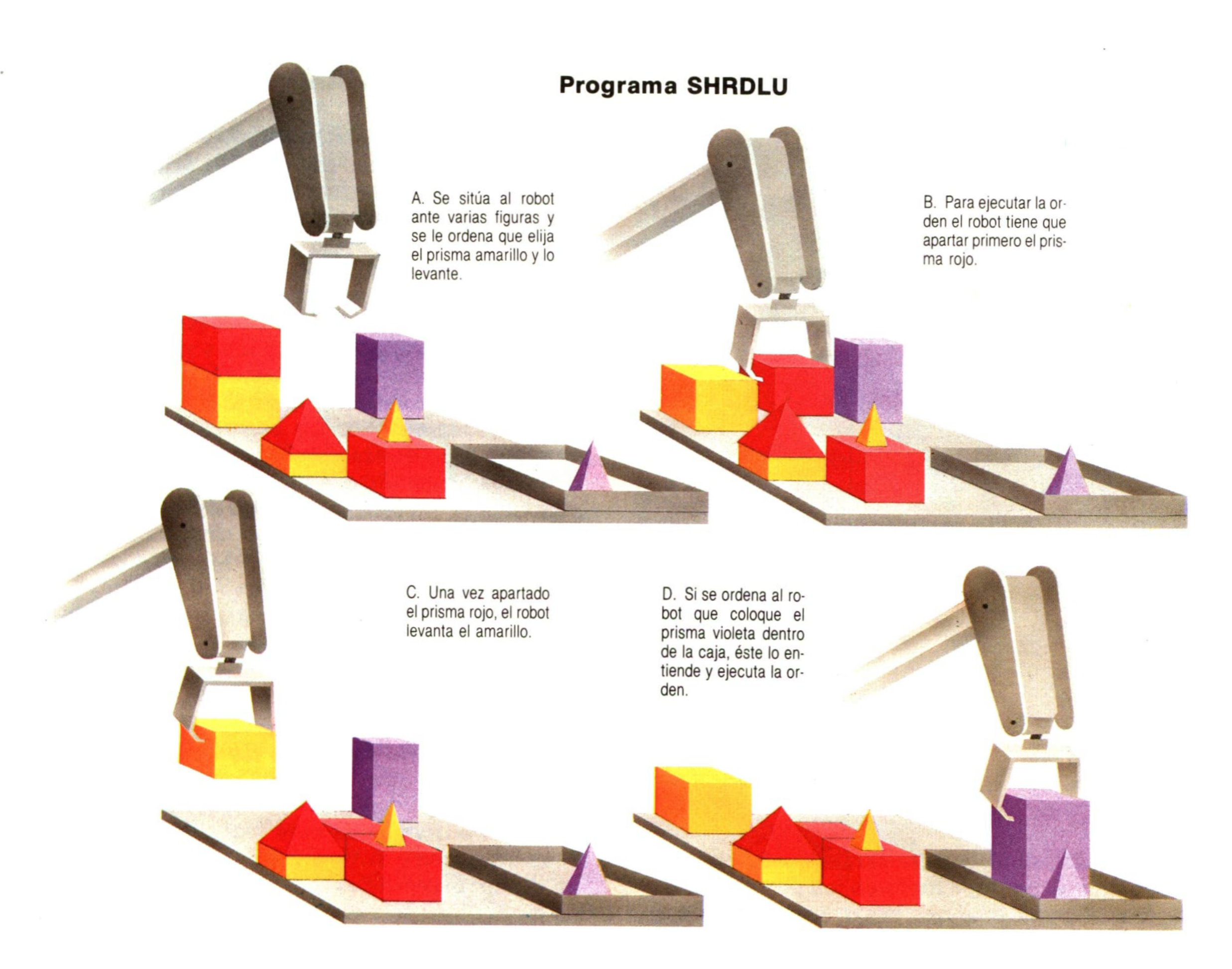
Bajo la pregunta de si las máquinas piensan o pueden pensar se cobija una dilatada historia de discusiones que no ha llegado a su fin y que, muy probablemente, perderá interés antes de llegar a una respuesta satisfactoria. Los más brillantes científicos han intervenido en la polémica para intentar sentenciar la cuestión. Turing, Von Neumann o Lucas son algunos de estos nombres famosos.

Tiempo atrás, considerar que las máquinas pudieran tener inteligencia parecía un absurdo, una estupidez infantil. Posteriormente, a medida que los progresos de la investigación cambiaban el panorama tecnológico, también cambió la atitud y se atribuyó un valor específico al problema teórico. Con ello se descubrió que la hipótesis de una inteligencia mecánica, artificial o simulada, abría nuevos interrogantes. El más serio de estos interrogantes se refería a la verdadera realidad de la inteligencia humana.

¿Qué rasgos fundamentales distinguen a los seres inteligentes y cómo operan biológicamente los procesos cognitivos? Esta nueva pregunta ha conducido a investigar una inadvertida laguna del saber humano. Con ello se ha visto que el ser humano, hasta el momento, se ha ocupado más de los resultados de su inteligencia que de los sutiles procesos y relaciones que la hacen posible. Estas relaciones y procesos atañen a la biología y a la lógica, lo que, en términos computacionales, puede traducirse como los ámbitos del hardware y el software.

¿SE PUEDE PRODUCIR ARTIFICIALMENTE LA INTELIGENCIA HUMANA?

Del ser humano se afirma su inteligencia porque posee intuición, inspiración, capacidad de organizar cadenas lógicas de pensamiento, sentimientos y expresión lingüística, entre otras cosas. El lenguaje es una manifestación externa de las otras capacidades o rasgos del conocimiento. No obstante, la definición resulta imprecisa y abstracta.



Programa SHRDLU. Este programa, elaborado por Terry Winograd de la Universidad de Stanford, permite a un robot dotado de visión reconocer objetos y colores, y realizar con ellos determinadas operaciones.

La palabra *inteligencia* procede del latín *intelligentia,* que significa la capacidad de entender o comprender. Esta etimología es poco iluminadora porque, en realidad, su origen se remonta a otro término latino, *legere,* que significa «coger» o «escoger». De ahí que *intelligere* comunique el significado de reunir elementos, escoger entre ellos y formar ideas, comprender, conocer.

De modo genérico, la actividad intelectiva agrupa, mediante un intrincado dispositivo neurológico, los procesos de la percepción, formación de impresiones, memorización, cotejo de imágenes, elección y gradación de éstas, comprensión y conocimiento. No es absurdo pensar que una máquina de extraordinaria perfección alcance a realizar estas tareas. También puede entenderse que el objetivo de una máquina pensante se circunscriba a ámbitos más lógicos que creativos, o emotivos, si parece remota una creación completa por medios artificiales de inteligencia.

Resulta difícil hacer una síntesis de la profusa polémica entre los que creen y los que no creen en la posibilidad de producir una inteligencia artificial. La breve exposición de sus respectivos argumentos arroja la luz suficiente acerca de sus enfrentadas posiciones y despliega un plano teórico que culmina un asombroso edificio de trabajos y experiencias desde los años cincuenta.

Unos y otros, tanto los que argumentan a favor como en contra, parten de unos presupuestos comunes que recogen los distintos ámbitos en que se fundamenta y manifiesta la inteligencia:

— percepción;
— asociación;
— memoria;
— imaginación o creatividad;
— razón;
— conciencia.

Sentadas estas capacidades —no menos abstractas y elusivas que la cuestión que se intenta dilucidar—, los argumentos contrarios a la inteligencia artificial se pueden resumir en los siguientes puntos:

- Las máquinas carecen de creatividad.
- Las máquinas no disponen de conciencia.
- Las máquinas no pueden alcanzar unos principios éticos con los que regir su conducta.

Frente a estos razonamientos negativos, los especialistas que creen en la legitimidad de la inteligencia artificial responden en los siguientes términos:

- Sí se produce el aprendizaje de las máquinas y se sientan las bases de la creatividad.
- El estadio de conciencia y la eticidad no son absolutamente imprescindibles para la afirmación de la inteligencia y, posiblemente, puedan conquistarse.

Sea como fuere, no conviene dejarse prender de la literalidad de la discusión. En el siglo XVII, Descartes asentó la tesis de que lo único que no funciona mecánicamente en el universo es nuestra capacidad de pensar. El ilustre racionalista francés afirmó el mecanicismo de la materia y la creatividad del pensamiento. En el presente siglo, no obstante, se ha demostrado que ello no es así mediante el uso de la computadora digital. La computadora es capaz de operar simulando el funcionamiento del cerebro y realizando así mismo con mucha mayor rapidez y precisión al menos algunas de sus actividades hasta ahora privativas de él.

¿Creatividad o mecanicismo?

Cuando se habla de la creatividad o del principio creativo se está admitiendo un salto cualitativo con respecto al resto de los procesos mecánicos. Es evidente que el pensamiento entraña una dificultad de análisis muy seria; pero ello no quiere decir que necesariamente haya de escapar a un modelo de compleja causalidad para su estudio.

Respecto a la conciencia de las máquinas, su carencia no impide su funcionamiento inteligente ni tampoco es la prueba de que no se pueda alcanzar la autoconciencia más adelante. Por supuesto, el conocimiento que discierne entre las cosas y el sujeto que conoce es superior al que sólo conoce las cosas. En el primero se da la conciencia. No obstante, desde un punto de vista histórico, el ser humano ha ganado la conciencia después de deambular durante períodos dilatados por entre las cosas. Y su andadura inteligente ya se había iniciado con anterioridad.

La vieja controversia sobre si es posible o no la inteligencia artificial ofrece un vivo campo para la dialéctica. Pero también se nutre de algo más que argumentos. Las actitudes emotivas provocan torrentes de palabras, sin atender a lo que en realidad está ocurriendo. Ello puede ser la explicación de que la discusión se agote a medida que deviene desfasada.

Lo cierto es que, paulatinamente, las computadoras están aprendiendo a ocuparse de una gran diversidad de tareas y que los sistemas expertos en curso demuestran capacidad de aprender y afinar en su actividad.

LA INTELIGENCIA ARTIFICIAL

La inteligencia artificial generalmente se expresa mediante la abreviatura I.A. Bajo esta denominación se recogen las realizaciones y los proyectos de la ingeniería del conocimiento. Si el nombre parece pretencioso, puede tomarse como la forma nominal para designar aparatos y sistemas tangibles, reales.

La I.A. tiene recorrido un largo trecho que se inicia a mediados de los años cincuenta. El elemento de arranque lo constituye la fabricación de las primeras computadoras electrónicas en la década anterior. La invención de la computadora hizo posible el viejo ideal de los seres humanos: *crear inteligencia, disponer almacenes de información y construir máquinas capaces de tratarla y de elaborar conocimientos.*

Etapas y campos de la inteligencia artificial

A pesar del empuje con que nació la I.A., las etapas por las cuales ha ido evolucionando no han resultado fáciles ni rápidas.

Históricamente se ha relacionado la aparición de la ingeniería del conocimiento con los sistemas automáticos para jugar a las damas y al ajedrez, entre otros juegos. El titubeante juego de los primeros avances de este tipo ha dado paso, finalmente, a versiones que han alcanzado la categoría de maestros. En el origen se hallan los primeros programas heurísticos. Entre ellos destaca el «Logic Theorist» (LT), escrito en 1956. El LT consistía en un novedoso programa de resolución general de problemas bien definidos, como el de *Las torres de Hanoi, El lobo y los corderos,* etc. De él se generan aplicaciones en psicología experimental relativas a la teoría de la resolución de problemas.

En la actualidad, la expresión «inteligencia artificial» todavía resulta opaca para el público. No manifiesta realidades concretas en las que está operando positivamente y, a la vez, resulta demasiado amplia para ser asimilada de un golpe. La inteligencia artificial recoge en su seno los siguientes aspectos fundamentales:

- — sistemas expertos;
- — robots;
- — procesamiento de lenguaje natural;
- — modelos de conocimiento;
- — visión artificial.

Cada uno de los campos señalados responde a sus propios objetivos. En ellos se aprecia la distribución de tareas y la investigación y realización especializadas. Es común a todos ellos el objetivo general de la inteligencia artificial. Y, así como existen etapas de profundización y dispersión, los progresos de cada campo revierten en los otros y en una previsible reunión integral de estos avances. La especializa-

ción puede ser un método que responde a una necesidad transitoria: posteriormente, podrá ser tan sólo una forma de rentabilización de acuerdo a las aplicaciones.

Sistemas expertos

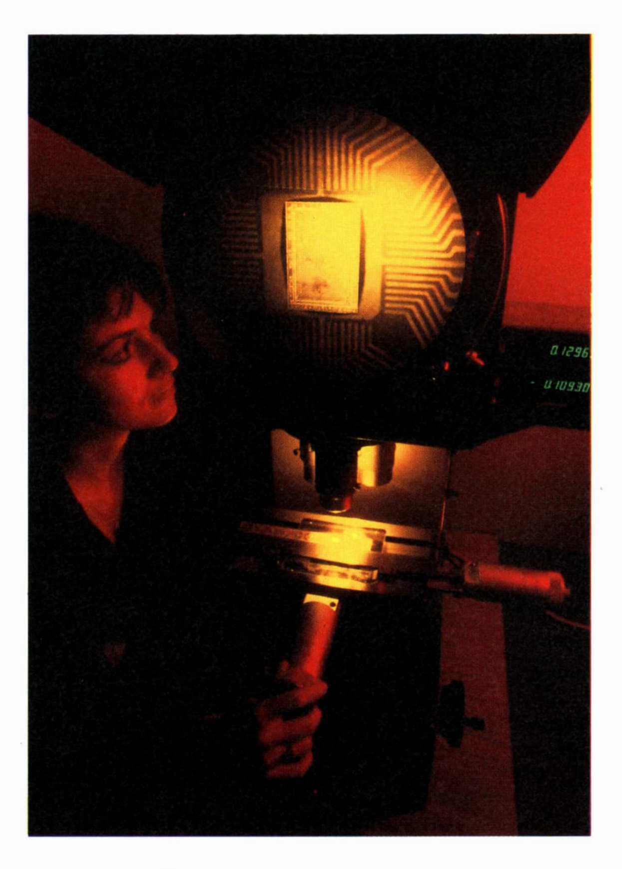
Los sistemas expertos o especializados constituyen una instrumentalización de la I.A. apasionante y muy útil. Son sistemas que acumulan saber perfectamente estructurado, de tal manera que sea posible obtenerlo gradualmente según las situaciones. Aquí desaparece el concepto de información en favor del de saber. Un sistema experto no es una biblioteca (que aporta información), sino un consejero o especialista en una materia (de ahí que aporte saber, consejo experimentado).

Un sistema experto es un sofisticado programa de computadora. Posee en su memoria y en su estructura una amplia cantidad de saber y, sobre todo, de estrategias para depurarlo y ofrecerlo según los requerimientos. Ello convierte al sistema (software-hardware) en un especialista en la materia para lo que está programado; se utiliza como apoyo o elemento de consulta para investigadores, médicos, abogados, geólogos, y otros profesionales. En la actualidad existe un gran número de sistemas expertos repartidos entre los campos más activos de la investigación y de la profesionalidad.

Veamos un ejemplo. Se trata de INTERNIST, un sistema experto en medicina. La medicina atrae, por el momento, buena parte de la atención de los diseñadores de sistemas y cuenta con el mayor número de programas. El sistema INTERNIST contempla el diagnóstico de las enfermedades de medicina interna u hospitalaria. Fue desarrollado en la universidad norteamericana de Pittsburg en 1977.

En los centros medicos que disponen de dicho sistema, el médico acude a la cónsola de la computadora después de haber reconocido al paciente y haber realizado los análisis que cree pertinentes. Entonces la máquina solicita al médico información sobre el paciente, y se establece una conversación a través de la pantalla y el teclado, similar a la que se establecería entre un médico y un reputado especialista al que se acude para contrastar un diagnóstico.

La computadora recibe el historial médico del enfermo, los síntomas y los resultados de pruebas y análisis. Con esta información, el sistema experto relaciona los datos de forma muy elaborada y comienza por desechar posibles diagnósticos hasta que llega a los que parecen más probables. Finalmente, elige uno y lo da a conocer con todo el detalle del proceso. Luego justifica su elección y el porqué de la posible enfermedad: cuadro clínico, historial, tratamiento, posibilidades de error, etcétera.



Una especialista examina los circuitos impresos en un chip con ayuda de un potente microscopio. Los procesos de fabricación del material computacional son quizás una de las aplicaciones más avanzadas de los sistemas expertos de inteligencia artificial. El control de calidad, sin embargo, aún se realiza con la intervención de la observación humana, aunque muchas fases son objeto de programas expertos.

La elaboración de los sistemas expertos exige el despliegue de un amplio equipo de ingenieros de I.A. y una larga tarea de organización del saber. El equipo trabaja con algún especialista en la materia de la aplicación; en el caso del INTERNIST, con brillantes y especializados médicos. Estos especialistas son denominados «informantes». La meta consiste en plasmar computacionalmente los «pasos» que el informante sigue para descartar unos diagnósticos y escoger el más acertado. Ello requiere pacientes sesiones para trasvasar el conocimiento del médico especialista a una programación que ha de incluir procedimientos de diagnóstico y conocimientos de enfermedades.

El lenguaje natural y otros ámbitos de la I.A.

Un gran objetivo, no carente de una abrumadora complejidad, se cifra en el *tratamiento del lenguaje natural*. Este objetivo consiste en

El sistema experto INTERNIST: un reputado especialista en diagnóstico

El INTERNIST cubre un amplio espectro de las enfermedades comunes en E.U.A.: unas 500 enfermedades. Ello supone la cobertura del 80 por ciento de las patologías de la medicina interna. Es evidente la ayuda que aporta a los médicos en caso de duda o, preferiblemente, para reafirmar su diagnóstico.
En el futuro, los sistemas expertos en medicina ofrecerán un inestimable servicio fuera de los hospitales, por ejemplo en los medios rurales y las expediciones espaciales, entre otras prometedoras posibilidades. En todo caso, la función de los sistemas expertos radica en asesorar al médico. Su servicio posee un carácter auxiliar, y al médico corresponde siempre la decisión.

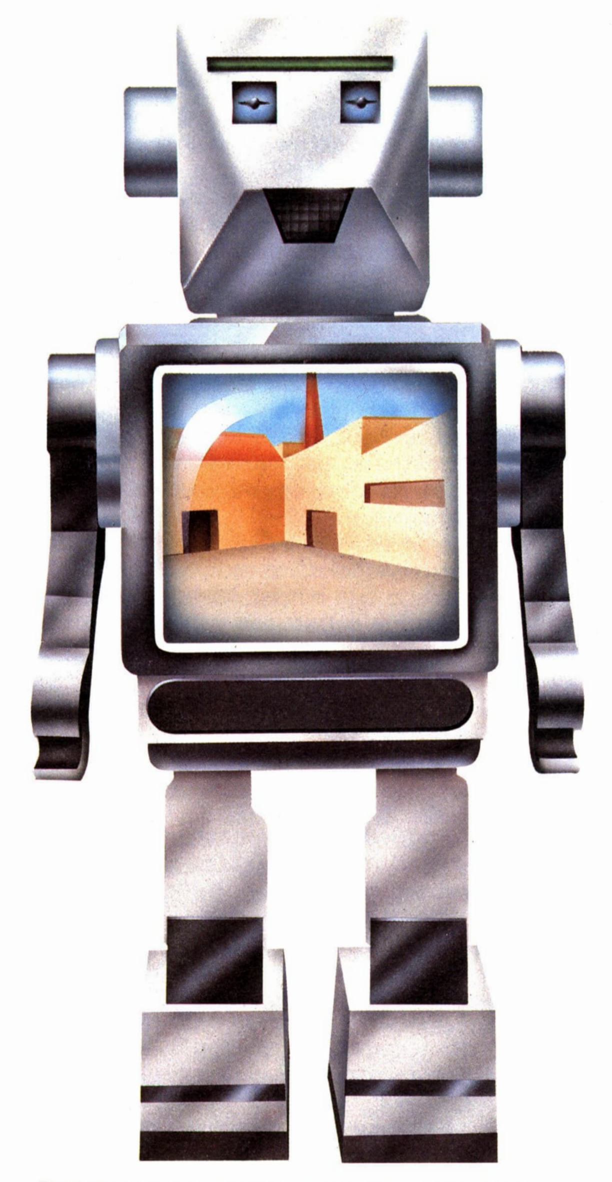
que las máquinas computacionales (y sus aplicaciones en robótica) puedan comunicarse con las personas sin ninguna dificultad de comprensión, ya sea oralmente o por escrito. Aquí encontramos la realización de un sueño largamente alimentado: hablar con las máquinas y que éstas entiendan nuestra lengua y, también, que se hagan entender en nuestra lengua.

La síntesis del lenguaje y el reconocimiento de voz forman dos aspectos del mismo propósito. Los logros que se han conseguido resultan a todas luces parciales e insuficientes, pero alentadores. Ciertas máquinas pueden interpretar las grafías de textos escritos y reproducirlos oralmente: leen los textos en voz alta.

Uno de los avances de este tipo (que sin duda representará una evolución notable en poco tiempo) es el programa FRUMP, elaborado en la universidad de Yale. FRUMP es capaz de leer historias cortas y resumirlas escuetamente con una gramática correcta y una expresión convincente.

Para lograr que las computadoras comprendan la lengua en la que la persona se expresa, es preciso pasar por una dilatada cadena de investigaciones en el campo de la acústica y ondas de lenguaje, análisis fonético y articulación, reglas de formación de frases o análisis sintáctico, el dominio semántico o de los conceptos y, finalmente, el análisis global de los actos de comunicación o pragmática.

El tratamiento del lenguaje natural tendrá una repercusión difícilmente imaginable. Su aplicación se extenderá al hasta ahora cegado camino de la traducción automática del lenguaje. Los textos o expresiones de una lengua se vertirán de manera automática e inmediata a otra lengua o idioma. Se prevé alcanzar mecánicamente un 90 por ciento de precisión, y el resto del material de traducción será tratado por personal especializado.



El dibujo muestra un robot, muchas veces imaginado y representado en los juguetes de los niños de hoy día, pero cuya imagen quizá sea corriente en un futuro no muy lejano, en que muchos niños puedan contar con un pequeño robot como compañero de juegos. El robot en cuestión conversará probablemente con ellos, inventará juegos, les ayudará en las tareas escolares y les permitirá comunicarse con otros amigos.

Junto al lenguaje, aparece otro objetivo capital, consistente en el *tratamiento de gráficos* y la *visión artificial.* De este planteamiento se desgranan aspectos como los de la percepción electrónica, selección y almacenamiento de

imágenes, reconocimiento visual de formas y objetos, producción de imágenes y diseño industrial (CAD/CAM), etcétera.

En último lugar, simplemente por el hecho de permitir la reunión de progresos de los otros objetivos, se encuentra la robótica.

Elementos de la I.A.

En verdad, la inteligencia artificial consiste en la asimilación de los procesos inductivos y deductivos del cerebro humano. Este intento de imitación se enfrenta a duras restricciones del hardware. Una computadora no es un cerebro; su complejidad electrónica se encuentra a una distancia abismal de la superior complejidad neurológica de aquél. La inteligencia artificial acepta el reto de la imitación de los procesos del cerebro aplicando mucho ingenio para aprovechar los medios de que se dispone y que se elaboran.

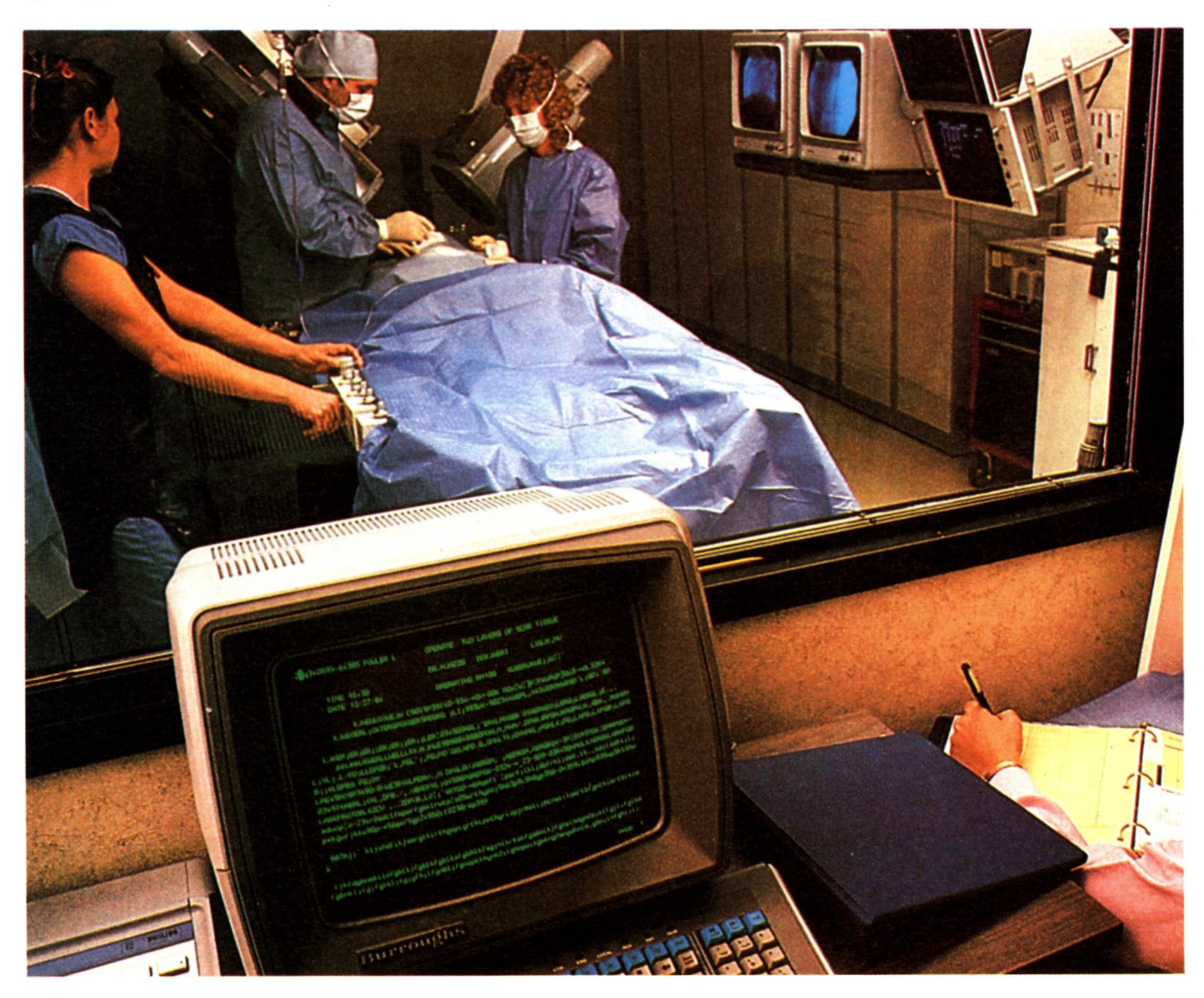
Sea cual sea la aplicación de que se trate, la I.A. se sustenta sobre los dos elementos siguientes:

- Estrategias de comportamiento inteligente.
- Saber o saberes.

Como se podrá apreciar, estos elementos forman una construcción coherente, son forma y contenido, o estructura y materia.

El primer elemento es el de las estrategias de comportamiento inteligente; se conjuga en la disposición de reglas para formular buenas inferencias o conjeturas y, también, en su uso para la búsqueda de una solución a la cuestión o tarea planteada. De esta forma, las estrategias son la parte estructural o formal.

Por oposición, el segundo elemento significa lo material o el contenido, y, por tanto, varía en cada caso de un modo más profundo; se trata del saber. En realidad, no se puede pretender reunir el saber, sino los saberes. Por ejemplo, cada sistema experto posee en memoria todos los conocimientos distintivos que tendría un es-



Intervención quirúrgica controlada por scanner y utilizando el INTERNIST, sistema experto para el diagnóstico instalado en el Memorial Medical Center de Jacksonville (Florida).

pecialista en la materia, sea un médico, un abogado o un químico. El saber que se recoge tiene un carácter especializado y alcanza un volumen conceptual considerable.

La estructura que presenta un sistema de información inteligente consta de tres niveles perfectamente integrados en una superarquitectura microelectrónica. Son tres niveles que cubren desde la relación exterior hasta la profunda organización interior. Éstos son:

• **Nivel externo.** Sirve para relacionar a la máquina con el medio y el ser humano. Este nivel está integrado por el *tratamiento del lenguaje natural* y el *tratamiento de las imágenes.* Con estos instrumentos la máquina percibe inteligentemente las señales que se le envían sin codificación especial, y adquiere un conocimiento.

• **Nivel medio.** En él se halla el sistema de resolución de problemas. La instrumentalización de esa capacidad se realiza mediante los *sistemas expertos,* que se configuran merced a unas estrategias de operación y una base de conocimientos orgánicamente ralacionados.

• **Nivel profundo.** Este último nivel corre paralelo a las funciones más profundas del cerebro. En él se sitúa, como proyecto, la capacidad de «aprender» automáticamente de la máquina. Tal proceso se concibe como la interpretación de diversas experiencias y su organización adecuada para ser utilizada en su caso. Finalmente, el nivel profundo está constituido por la base de conocimientos generales y la flexibilidad para ampliarse por sí misma.

Este diseño en tres niveles muestra la estructura de un sistema de inteligencia artificial y refleja los progresos que se tienen que alcanzar con la quinta generación de computadoras. Necesariamente, I.A. y quinta generación se asocian en lo que es un salto cuantitativo y, sobre todo, cualitativo del tratamiento de la información.

Un técnico trabajando con un sofisticado robot de fabricación suiza para la lectura de textos.



LA QUINTA GENERACIÓN DE COMPUTADORAS

Las grandes etapas de la computación se conocen con el nombre de generaciones. Se comienza a contar las generaciones que se suceden a partir de la comercialización de las computadoras, en los años cincuenta. La etapa de experimentación previa se considera anterior a cualquier generación.

Las generaciones que se han sucedido son cuatro, y dentro de cada una de ellas pueden distinguirse distintas fases. Cada generación ha significado un avance considerable en los aparatos y la programación.

En abril de 1982 se puso en marcha un ambicioso plan japonés para dar lugar a la quinta generación. La previsión de este plan habla de unos diez años de trabajos para lograr el objetivo inicial. De manera paralela, Japón cuenta con otros dos planes: el Proyecto nacional de la computadora ultrarrápida y el Proyecto nacional de robótica, que complementa el primero, y confluyen ambos en macroobjetivos comunes. En otros países se desarrollan tareas similares.

Las cuatro generaciones que hasta el momento se han sucedido se caracterizan por la simple oposición distintiva de su tecnología fundamental:

— primera generación: válvula de vacío;
— segunda generación: transistor;
— tercera generación: circuito integrado;
— cuarta generación: microcircuito integrado.

Por supuesto, bajo esta lineal definición existen otras características realmente determinantes. Lo que se espera que represente la quinta generación no se cifra en la simple modificación de algún elemento del hardware o la circuitería. Si las previsiones se cumplen, esta nueva generación aportará un cambio radical que contrastará con las anteriores generaciones consideradas como un solo bloque.

Las computadoras que se alinean en el tiempo que va de la primera a la cuarta generación responden al modelo de máquina VON NEUMANN (en honor al matemático que la diseñó), que funciona paso a paso, siguiendo una secuencia o proceso lineal. Con la quinta generación se pretende multiplicar la velocidad, disponer procesamientos paralelos, diseñar una arquitectura de hardware-software muy superior y utilizar el lenguaje natural.

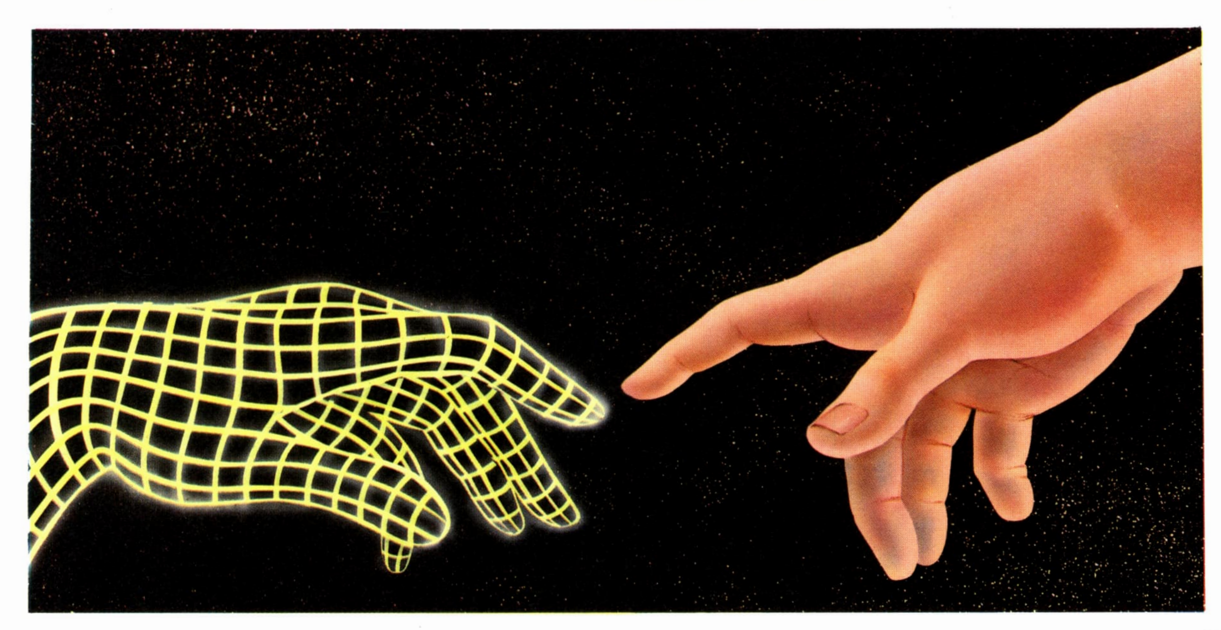
La renovación que aporta la quinta generación es múltiple. Ya no viene determinada úni-

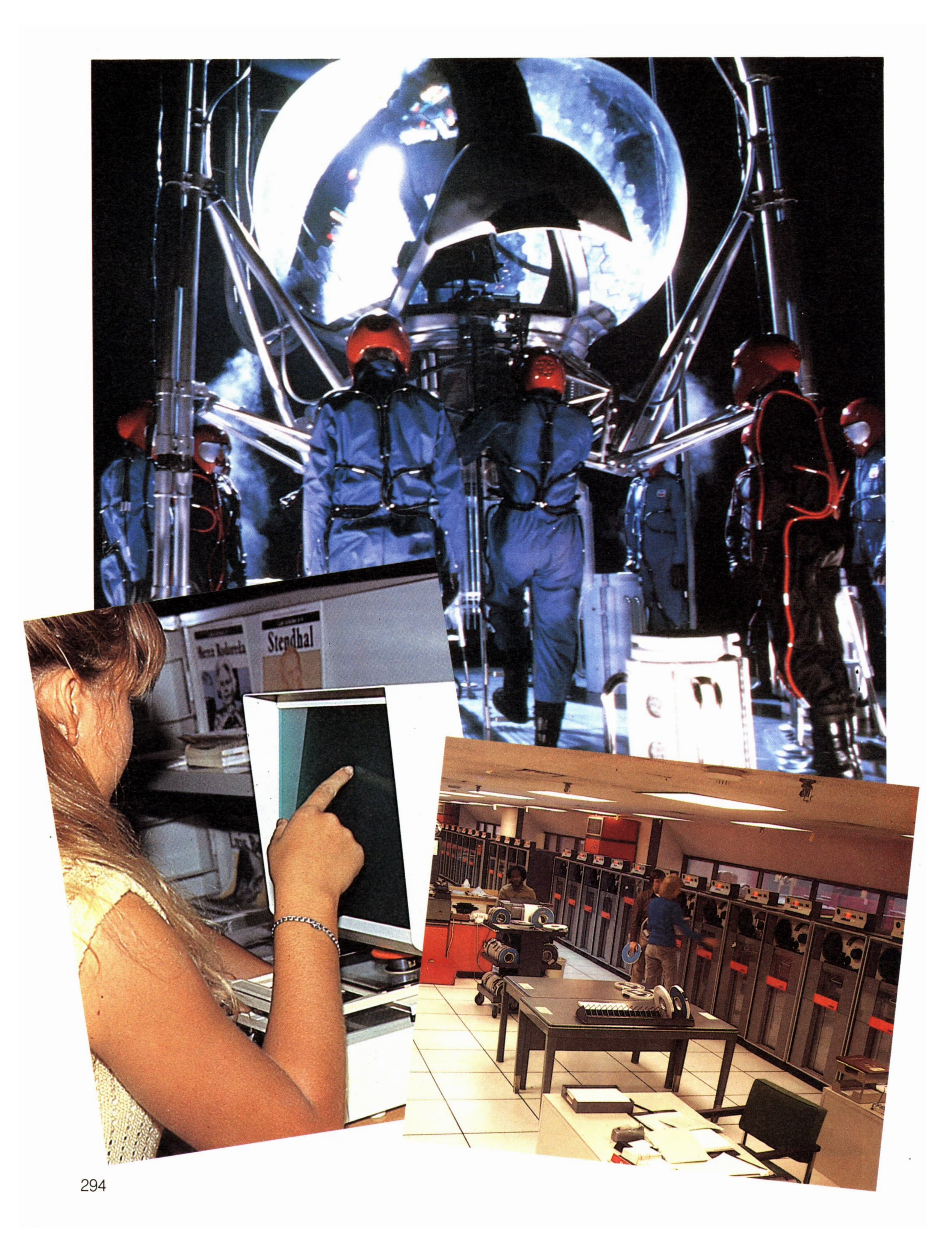
La quinta generación frente a las generaciones anteriores		
Máquina de Von Neumann (proceso lineal)	Primera generación:	válvula (1951-1958)
	Segunda generación:	transistor (1959-1964)
	Tercera generación:	circuito integrado (1965-1970)
	Cuarta generación:	microcircuito integrado (1971-?)
Máquina de arquitecturas no Von-Neumann (procesos paralelos e interconectados)	Quinta generación:	remodelación hardware-software • Mayor velocidad (de 1.000 a 10.000 veces superior a la de la 4.ª generación) • Mayor miniaturización de elementos • Aumento de la capacidad de memoria • Diversas memorias • Multiprocesador (procesadores interconectados) • Lenguaje natural • Elaboración inteligente del saber y no mero tratamiento de datos

camente por la introducción de cambios materiales o de hardware, sino por la combinación de novedades en el material y en la programación. La remodelación es aún más amplia, ya que contempla conjuntamente los dos aspectos de la computación, el hardware y el software (que se apoyan y se oponen como las dos caras de una moneda).

Otra característica que merece una atención especial se refiere a la capacidad de su tarea. El tipo de trabajo propio de las máquinas de las generaciones precedentes se resume en el tratamiento de datos. Esta gestión computacional se quiebra y puede alcanzar un grado superior con los planes de la quinta generación. El grado superior consiste en la adquisición de información y, a partir de los materiales y estructuras de que se dispone, en la elaboración de conocimientos, es decir, el despliegue de un tratamiento inteligente.

Hace tiempo que la computación ha superado el limitado ámbito del cálculo. Las computadoras no sólo calculan, sino que se mueven con comodidad por un mundo interior de símbolos. El ámbito se ampliará mucho más, y la computación roturará nuevos límites, más allá de lo conocido hasta hoy. El futuro inminente que se anuncia es el del aprovechamiento de la computadora y los sistemas de inteligencia artificial para adquirir artificialmente conocimientos. A partir de entonces, las máquinas ya no sólo aportarán fuerza o habilidad (como lo hacen todos los ingenios que reproducen mecánicamente el brazo y la mano humanas), sino que también proveerán al hombre de un conocimiento artificial del mundo.




Merce Rodoreda
Stendhal

Instrucciones más comunes del lenguaje BASIC

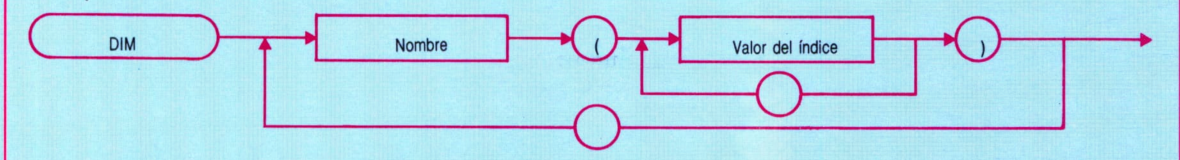
Para describir las principales instrucciones del lenguaje, lo haremos agrupándolas según su función e iremos explicando grupo a grupo. De esta manera empezaremos por las instrucciones de definición de matrices.

INSTRUCCIONES DE DEFINICIÓN DE MATRICES

El BASIC, al igual que muchos otros lenguajes de programación, permite trabajar con vectores o matrices; por lo tanto, posee instrucciones para ello.
Antes de asignar valores a los elementos de un vector o matriz, éstas se deben definir para que el BASIC guarde el espacio correspondiente de memoria; esto se hace mediante unas instrucciones específicas que detallaremos a continuación. En este apartado hablaremos siempre de matrices, ya que un vector es una matriz de una sola dimensión. Si no se recuerda cómo se representan las matrices, aconsejamos leer el apartado «Variables con índice» de este mismo volumen.

La instrucción DIM

Mediante esta instrucción se define el nombre de una matriz o vector, el número de dimensiones de que constará y el valor máximo del índice de cada dimensión. Además, con una misma instrucción DIM, pueden definirse más de una matriz o vector. Las matrices pueden ser numéricas o alfanuméricas.
Su formato es el siguiente:



DIM — Nombre de la instrucción.
Nombre — Nombre que se le va a dar a la matriz.
Valor índice — Máximo valor que tomará cada uno de los índices.

Si seguimos atentamente la figura, podemos ver que puede tener varios formatos según estemos definiendo una sola matriz o varias y según la matriz que estamos definiendo sea de una sola dimensión o de varias.

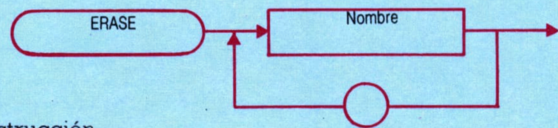
Ejemplos de todos los casos

20 DIM M(5)	Se define una matriz numérica de una sola dimensión, cuyo nombre es A y que constará de 5 elementos.
30 DIM MA(5,25)	Se define una matriz numérica de dos dimensiones, cuyo nombre es MA y que constará de 125 elementos (5×25). Podemos expresarlo también como 5 vectores de 25 elementos.
40 DIM M(5), MA(5,25)	Se define en una sola instrucción DIM lo que antes hemos hecho en dos.
20 DIM M$(5)	Se define una matriz alfanumérica de una sola dimensión cuyo nombre es M$ y que constará de 5 elementos.
30 DIM MA$(5,25)	Se define una matriz alfanumérica de dos dimensiones cuyo nombre es MA$ y que constará de 125 elementos (5×25).
40 DIM M$(5), MA$(5,25)	Se define en una sola instrucción DIM lo que hemos hecho en dos.
50 DIM M(5), MA(5,25), M$(5), MA$(5,25)	Con esta sola instrucción DIM, conseguimos definir todas las matrices que antes hemos definido una a una o dos a dos.

La instrucción ERASE

Así como una matriz tiene que definirse para que ocupe espacio en memoria, también, cuando ya no se utiliza para nada, puede anularse.

Supongamos que hemos adquirido una matriz de 25×25 elementos, y hemos estado trabajando con ella en un programa BASIC, pero ha llegado un momento en que ya no se va a usar más (dentro del mismo programa). Entonces, el espacio que ocupa en memoria puede liberarse mediante la instrucción ERASE (dentro del mismo programa), para poder definir más matrices o simplemente para disponer de más memoria libre.
Su formato es el siguiente:



ERASE — Nombre de la instrucción.
Nombre — Nombre de la matriz que se quiere anular.

Ejemplos de todos los casos

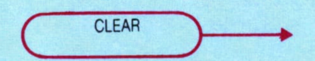
10 ERASE A	Liberará el espacio de memoria ocupado por la matriz cuyo nombre es A, en el caso de que se haya definido anteriormente, mediante la instrucción DIM.
20 ERASE A,B$,B	Liberará el espacio ocupado por las matrices cuyo nombre es A, B$ y B.
10 DIM M(5), MA(5,25), M$(5), MA$(5,25) ⋮	Definimos estas cuatro matrices.
200 ERASE MA(5,25), MA$(5,25)	Anulamos dos de ellas, con lo que liberamos el espacio que ocupaban.
210 DIM VE(100), VI$(10,10)	Volveremos a definir otras dos, con lo que volvemos a ocupar tal espacio.

INSTRUCCIONES DE ASIGNACIÓN

En un lenguaje de programación es fundamental el uso de variables; en el BASIC también. Las variables, como ya se ha explicado, pueden cambiar su valor frecuentemente, pero esta variación se debe producir a través de instrucciones BASIC.
Estas instrucciones sirven también para asignar un primer valor a unas variables BASIC, que no sea el cero o nulo, o para asignar a todas el valor cero o nulo.
Estas instrucciones son las siguientes:

La instrucción CLEAR

La función de esta instrucción es la de poner a cero todas las variables numéricas y a valor nulo todas las alfanuméricas. Normalmente, siempre se encuentra en las primeras líneas de programa.
Su formato es el siguiente:



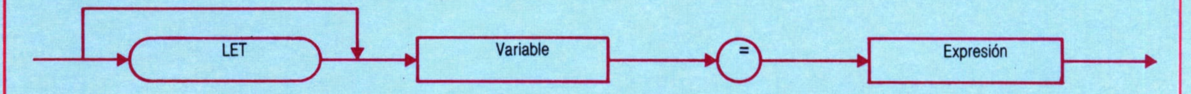
CLEAR — Nombre de la instrucción.

Ejemplo

20 CLEAR Pone a cero todas las variables numéricas y a valor nulo todas las alfanuméricas.

La instrucción LET

Es la instrucción que suele ser más utilizada dentro de un programa escrito en BASIC. Su función es la de asignar un valor a una variable, ya sea numérica o alfanumérica.
Su formato es el siguiente:



LET — Nombre de la instrucción, que puede omitirse, como vemos en la figura.
Variable — Nombre de la variable numérica o alfanumérica a la que se le quiere asignar un valor.
Expresión — Puede ser una expresión numérica o alfanumérica, pero siempre debe ser numérica

cuando el parámetro «variable» sea una variable alfanumérica. Nunca se puede realizar una asignación del valor de una expresión alfanumérica a una variable numérica, ni viceversa.

Ejemplos de todos los casos

10 LET A=2	Se le asigna el valor 2 a la variable numérica A. A partir de ese momento la variable A tendrá valor 2.
20 LET D=(B+C)/2	Suponiendo que B sea igual a 5 y C sea 9, el valor de la expresión (B+C)/2 será (5+9)/2, o sea 7. Entonces D pasará a tener valor 7.
30 LET A$="MARC"	Se le asigna el valor "MARC" a la variable alfanumérica A$.
40 LET D$=B$+C$	Suponiendo que el valor de B$ sea "MARC" y el de C$ sea "SERRA", entonces D$ tomará el valor "MARC SERRA".
10 A=2 20 D=(B+C)/2 30 A$="MARC" 40 D$=B$+C$	Estas cuatro sentencias son equivalentes a las cuatro anteriores, ya que se puede omitir el nombre de la instrucción LET sin que ello afecte a la función de la instrucción.

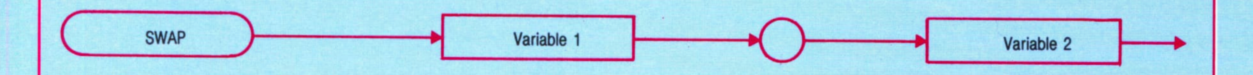
Nunca será aceptable al programar en BASIC que se haga una doble asignación en una misma instrucción. Por ejemplo:

```
10 LET A=B=10
```

Esta instrucción no será correcta y producirá un error de sintaxis.

La instrucción SWAP

Esta instrucción permite intercambiar los valores de dos variables, es decir, asignar a la primera el valor de la segunda y a ésta el valor de la primera.
Su formato es el siguiente:



SWAP — Nombre de la instrucción.
Variable 1 — Nombre de la primera variable.
Variable 2 — Nombre de la segunda variable.

Ejemplo

10 SWAP A, B — Las variables A y B intercambiarán sus valores.
20 SWAP A$, B$ — Igualmente pasará con los valores de A$ y B$.

Si esta instrucción no existiera, se podría realizar lo mismo, pero con tres instrucciones en lugar de una. Por ejemplo, supongamos el pequeño programa siguiente:

```
10 LET A=5
20 LET B=9
30 PRINT A;B
40 SWAP A,B
50 PRINT A;B
60 END
```

Vemos que en la línea 10 se asigna a la variable A el valor 5 y, en la 20, el valor 9 a la variable B. En la línea 30 se imprimen los valores de A y de B, que serán **5 y 9**. En la siguiente línea, la 40, se intercambian los valores entre A y B, de tal manera que en la línea 50 se imprimen los valores de A y B de nuevo; esta vez la impresión será **9 y 5**, lo cual quiere decir que han cambiado su valor.

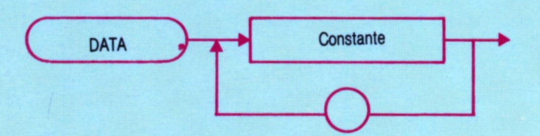
Este mismo programa se construiría sin usar la instrucción SWAP, pero deberíamos escribir más instrucciones. Ejemplo:

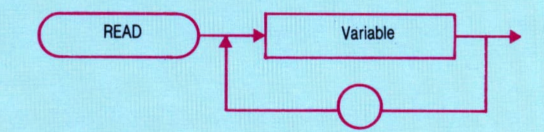
```
10 LET A=5
20 LET B=9
30 PRINT A;B
40 LET C=A
50 LET A=B
60 LET B=C
70 PRINT A;B
80 END
```

En este programa, hasta la línea 30 hacemos lo mismo que en el anterior y, por lo tanto, al ejecutar la línea 30 se imprimirá | 5 y 9 |; sin embargo, en la 40, 50 y 60 realizamos tres asignaciones que equivalen a la instrucción 40 del primer programa. Veamos qué hacemos: primero asignamos el valor de A a C, con lo cual C pasará a valer 5; asignamos después el valor de B a A, con lo cual A, en lugar de 5, ahora vale 9; por último, asignamos a B el valor de C, con lo cual B pasará de valer 9 a valer 5 y, cuando se ejecute la línea 70, se imprimirá | 9 y 5 |, que será lo contrario a la anterior impresión.

LAS INSTRUCCIONES DATA/READ

Estas dos instrucciones se complementan. El hecho de especificar instrucciones DATA en un programa BASIC hace que se cree un fichero interno con los datos de tales instrucciones, datos que posteriormente pueden leerse y asignarse a variables mediante instrucciones READ.
El formato de tales instrucciones es:





DATA — Nombre de la instrucción.
Constante — Cada uno de los valores que se almacenará en el fichero interno que se creará.

READ — Nombre de la instrucción.
Variable — Nombre de cada una de las variables donde se irán asignando las constantes almacenadas en el fichero interno.

Ejemplos

Supongamos el programa siguiente:

```
10 READ A,B,C,D
20 DATA 5,10,15,20
30 PRINT A,B,C,D
```

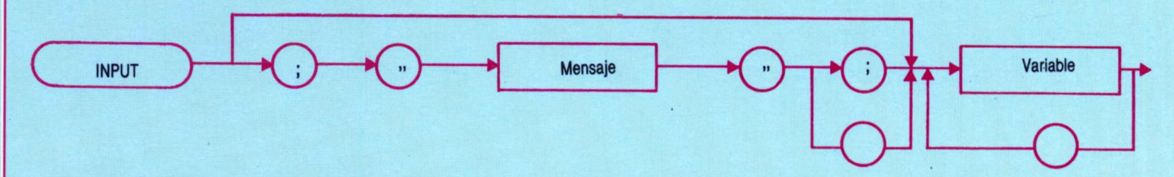
En la variable A se almacenará el valor 5, en la B el 10, en la C el 15 y en la D el 20; por lo tanto, al ejecutar la instrucción de la línea 30, se imprimirán los valores 5, 10, 15, 20.
La instrucción DATA no tiene necesidad de ir delante de la instrucción READ; de hecho, en el programa del ejemplo no está así.

INSTRUCCIONES DE ENTRADA DE DATOS

En todo lenguaje deben existir instrucciones mediante las cuales, al ejecutar un programa en un momento determinado, el operador pueda introducir una serie de datos por el teclado, datos que el programa necesita para realizar sus cálculos o tratamientos.
Existen dos instrucciones de este tipo, que son INPUT y LINE INPUT. Sin embargo, solamente describiremos la primera, que es la más utilizada.

La instrucción INPUT

La función de esta instrucción consiste en leer datos que se entran desde teclado y asignarlos a una o más variables especificadas en la instrucción.
Su formato es el siguiente:



INPUT — Nombre de la instrucción.
Mensaje — Literal que aparece en pantalla para avisar al operador que debe digitar algún dato. Puede omitirse.
Variable — Cada uno de los nombres de variable en las que se irán almacenando los datos entrados desde el teclado.

Podemos ver en esta figura que después de las segundas comillas (") se pueden elegir dos caminos: Uno escribiendo un (;) u otro escribiendo una (,). Cuando en ese lugar se especifica un punto y coma, siguiendo al mensaje en pantalla e inmediatamente antes del lugar en que el operador debe digitar el dato, aparecerá un signo de interrogación cerrado (?). Si se especifica una coma, no aparecerá.

Ejemplos de todos los casos

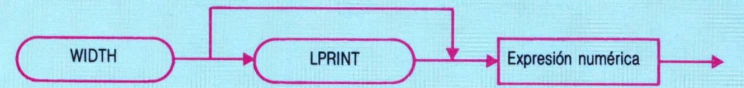
10 INPUT A	En la pantalla se visualizará un signo de interrogación (?) reclamando al operador la entrada de un dato. Tal entrada debe terminarse con la tecla [CR]. El dato entrado será almacenado en la variable A.
20 INPUT; "Lado"; B	En la pantalla se visualizará el mensaje "Lado" seguido del signo de interrogación (?) reclamando la entrada de un dato. Tal dato, una vez entrado, se almacenará en la variable B.
30 INPUT; "Lado", C	Hará lo mismo que la instrucción anterior; sin embargo, no aparecerá después del mensaje el signo de interrogación (?), porque antes de la variable C se ha especificado una coma (,) y no un punto y coma (;).
40 INPUT; "Datos"; A,B,C	En este caso se deberán entrar tres datos, que se irán almacenando a las variables especificadas en el mismo orden en que están las variables, es decir el 1.º en A, el 2.º en B y el 3.º en C. Es una forma de realizar varias entradas con una sola instrucción.

INSTRUCCIONES DE SALIDA DE DATOS

Los dos dispositivos más importantes de una computadora, en donde quedan reflejados los datos de salida de un programa, son la pantalla y la impresora. Estos datos son enviados a tales dispositivos mediante instrucciones que serán diferentes según el lenguaje que se utiliza. En BASIC también existen estas instrucciones, las más importantes de las cuales pasamos a describir:

La instrucción WIDTH

Esta instrucción, en realidad, no es de salida de datos, sino que se utiliza para preparar mejor dicha salida.
Define la amplitud máxima en número de caracteres de la línea de la pantalla o de la impresora, que podrá ser visualizada o impresa por todas las instrucciones de salida.
Su formato es el siguiente:



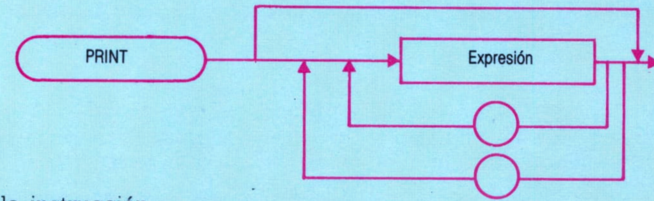
WIDTH — Nombre de la instrucción.
LPRINT — Forma parte también del nombre de la instrucción cuando ésta se refiere a la salida por impresora.
Expresión — Puede ser una constante numérica o una expresión con constantes y variables.

Ejemplos de todos los casos

10 WIDTH 80	Define la amplitud de la línea de visualización de la pantalla, de 80 caracteres de longitud.
20 WIDTH LPRINT 130	Define la amplitud de la línea de impresión de la impresora, de 130 caracteres de longitud.

La instrucción PRINT

Es la instrucción utilizada para visualizar información en la pantalla de la computadora.
Su formato es el siguiente:



PRINT — Nombre de la instrucción.
Expresión — Cualquier expresión, ya sea numérica o alfanumérica.

Ejemplos de todos los casos

10 PRINT	Se visualizará una línea en blanco en la pantalla.
20 PRINT "1"	Se visualizará un 1 en la pantalla.
40 PRINT "1"; "PEPE"	Se visualizarán los dos valores anteriores en la misma línea de la pantalla y uno detrás de otro.
50 PRINT "ÁREA =";A	Suponiendo que la variable A valga 50, se visualizará ÁREA=50; o sea que los literales entre comillas se visualizan tal como se especifican en la instrucción y de las variables se visualiza su valor.

```
10 A=5
20 B=10
30 C=A*B
40 PRINT "EL ÁREA ES";A;"×";B;" IGUAL A"; C
```

En este caso se visualizará en pantalla la siguiente línea:

EL ÁREA ES 5×10 IGUAL A 50

Notemos que el signo (;) sirve para separar las variables entre sí o para separar los literales de las variables.
En esta instrucción puede utilizarse también la palabra TAB, junto con una constante entre paréntesis, para indicar en qué posición de la línea se quiere visualizar la variable o el literal. Por ejemplo, la instrucción de la línea 40 del ejemplo anterior podría quedar así:

40 PRINT TAB(10); "EL ÁREA ES";A;"×";B;" IGUAL A";C

En este caso, tal mensaje, en lugar de visualizarse a partir de la primera posición de la línea, se visualizaría a partir de la 10.

La instrucción LPRINT

Esta instrucción realiza lo mismo que la instrucción PRINT, pero, en lugar de visualizar información en pantalla, la imprime en la impresora. Su formato es el mismo que el de la instrucción PRINT. En cualquier instrucción PRINT, al añadir una L delante del nombre de la instrucción, ésta ya no visualizará la información en la pantalla, sino que la imprimirá en impresora, si está conectada.
Por ejemplo, si a la instrucción

40 PRINT TAB(10); "EL ÁREA ES";A;"×";B;" IGUAL A"; C

le añadimos una L delante de PRINT, el resultado lo dará en impresora y no en pantalla como antes:

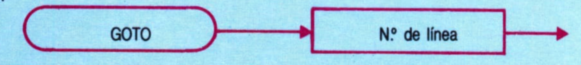
40 LPRINT TAB(10); "EL ÁREA ES";A;"×";B;" IGUAL A"; C

INSTRUCCIONES DE CONTROL

Normalmente, cuando se ejecuta un programa BASIC, sus instrucciones se ejecutan una a una secuencialmente, pero a veces es necesario saltar de una línea a otra diferente de la siguiente.
A este salto se le llama comúnmente cesión del control. En este lenguaje existen varias instrucciones que realizan tal cesión del control; no secuencialmente, sino a otra instrucción diferente a la siguiente, dependiendo o no del cumplimiento de una determinada condición, o a través de estructuras de bucle que se han visto en el capítulo anterior.
En este apartado, por lo tanto, describiremos las instrucciones de este tipo más usadas.

La instrucción GOTO

Esta instrucción cede el control de ejecución del programa a la línea especificada de una manera incondicional.
Su formato es el siguiente:



GOTO — Nombre de la instrucción.
Nº de línea — Número de la línea de programa a la que se le va a ceder el control de ejecución.

Ejemplos

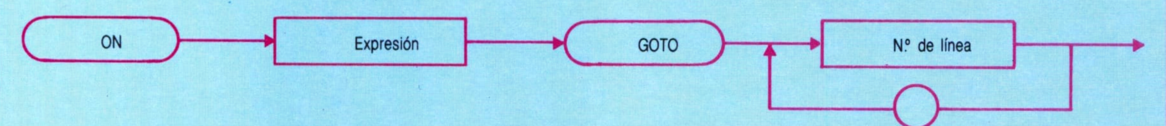
10 REM	
20 A=5	En este pequeño programa podemos ver que se irán ejecutando las instrucciones
30 B=10	secuenciales a partir de la línea de programa 10, hasta llegar a la línea 70, en la
40 PRINT A,B	cual encontramos una instrucción GOTO que cede el control de la ejecución a la
50 A=A+5	línea 40.
60 B=B+10	Por lo tanto, su funcionamiento será el siguiente:
70 GOTO 40	

línea 20 —Se le asigna a la variable A el valor 5.
línea 30 —Se le asigna a la variable B el valor 10.
línea 40 —Se visualiza en pantalla el valor de las dos variables A y B.
línea 50 —Se le asigna a la variable A el valor de ella misma más 5, con lo que pasará a valer 10.
línea 60 —Se le asigna a la variable B el valor de ella misma más 10, con lo que pasará a valer 20.
línea 70 —Se cede el control otra vez a la línea 40, con lo cual se volverá a ejecutar el programa a partir de la línea 40.

A partir de este momento el programa entra en un proceso sin salida, al ejecutarse las líneas 40, 50, 60, 70, 40, 50, 60, 70 y así sucesivamente. Es decir, este programa no terminará nunca.

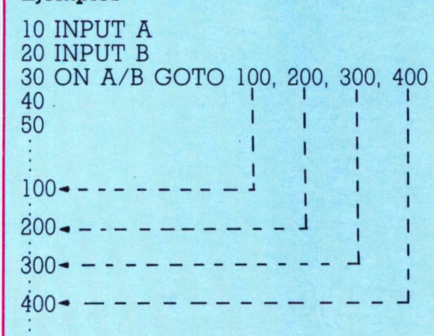
La instrucción ON... GOTO

Esta instrucción transfiere el control de la ejecución de un programa a una línea o a otra según el valor de una expresión especificada después de la palabra ON.
Su formato es el siguiente:



ON — Palabra que forma parte del nombre de la instrucción.
Expresión — Normalmente es una expresión numérica, de cuyo resultado depende que se ceda el control a una línea o a otra del programa.
GOTO — Palabra que forma parte del nombre de la instrucción.
N.º de línea — Cada uno de los números de línea de programa a la que se puede acceder el control de la ejecución de tal programa.

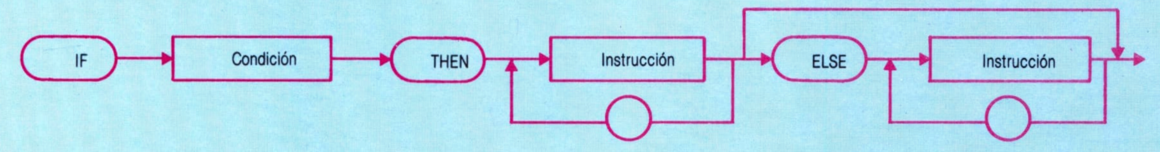
Ejemplos



En este programa podemos ver que en las instrucciones 10 y 20 se realiza la entrada de dos cantidades numéricas que se almacenan en las variables A y B. En la línea 30 se efectúa una bifurcación o cesión de control que depende del valor de la expresión A/B, de tal manera que, si A/B es igual a 1, el control se cederá a la línea 100; si es igual a 2, a la 200; si es igual a 3, a la 300 y si es igual a 4, a la 400; pero, si es menor de 1 o mayor de 4, la ejecución seguirá por la línea 40. Esta instrucción siempre se comporta así; es decir, cuando se ceda el control a una línea de las especificadas en la serie después del GOTO, será porque la posición de tal línea dentro de la serie es igual al valor de la expresión.

La instrucción IF ... THEN ... ELSE

Esta instrucción se usa muy a menudo para poder ceder el control a una determinada línea de programa o para realizar un tratamiento diferente de un proceso, según se cumpla o no una determinada condición.
Su formato es el siguiente:



IF — Palabra que forma parte del nombre de la instrucción.
Condición — Puede ser una expresión numérica, de comparación o lógica.

THEN — Palabra que forma parte del nombre de la instrucción.
Instrucción. Puede ser una instrucción cualquiera.
ELSE — Palabra que forma parte del nombre de la instrucción.
Instrucción. Puede ser una instrucción cualquiera.

Notemos que es en esta figura donde aparece por primera vez el signo (:), que se utiliza en BASIC para enlazar dos o más instrucciones dentro de una misma línea de programa. Sin embargo, lo explicaremos mejor más adelante.

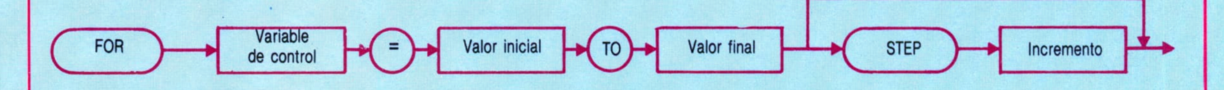
Ejemplos de todos los casos

```
10 INPUT A,B
20 IF A > 5 THEN GOTO 100
30 IF A < 4 THEN A=1: GOTO 40 ELSE A=2
40
50
:
:
100
:
110 IF B THEN GOTO 200
120
:
:
200
:
250 IF (A > 2 AND B > 3) THEN GOTO 400
260
:
:
400
```

En el caso de la línea 20, si el valor de la variable A es mayor de 5, se cederá el control a la línea 100 si no, se seguirá ejecutando la siguiente instrucción, es decir, la 30.
En el segundo caso de la línea 30, vemos que si el valor de la variable A es menor de 4, entonces se asigna valor 1 a tal variable y se cede el control a la línea 40. Si no es menor de 4, entonces se le asigna valor 2 y se sigue por la línea 40.
En el caso de la línea 110, vemos que no se hace una comparación, sino que solamente se expecifica una variable. Lo que ocurrirá es que el sistema investigará si el valor de la variable es igual a cero, con lo que la condición será falsa, o no es igual a cero, con lo que la condición será verdadera. Por lo tanto, si B es igual a cero, entonces se seguirá por la siguiente instrucción 120; si no, se cederá el control a la línea 200.
Por último, en la línea 250 podemos ver una condición compuesta de dos expresiones de comparación unidas por un operador lógico (AND). El resultado de esta condición viene dado por la tabla lógica que se encuentra en el apartado de expresiones lógicas. Es decir, si las dos comparaciones son verdaderas, la condición será verdadera; si una de las dos o las dos son falsas, la condición será falsa. Si es verdadera, se cederá el control de ejecución a la línea 400; si no, a la siguiente, la 260.

LAS INSTRUCCIONES FOR/NEXT

Veremos ahora dos instrucciones fundamentales en el BASIC, que se utilizan para formar bucles sencillos y bucles anidados. Es muy normal en cualquier programa BASIC realizar la construcción de uno o varios bucles; estas dos instrucciones que se complementan sirven para construirlos.
Formato de la instrucción FOR:

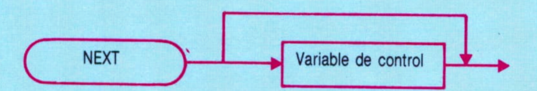


FOR — Palabra que forma parte del nombre de la instrucción.
Variable de control — Nombre de una variable entera o de simple precisión.
Valor inicial — Valor inicial que tomará la variable de control.
TO — Palabra que forma parte del nombre de la instrucción.
Valor final — Es el valor final que tomará la variable de control por encima del cual el bucle llegará a su final.

STEP — Palabra que puede formar parte del nombre de la instrucción o no; es decir, es opcional.

Incremento — También es opcional, pero, si se especifica, indica el valor que se le irá sumando al valor de la variable de control hasta llegar a su valor final.

Si no se especifica | STEP → incremento |, se toma por defecto el valor de incremento igual a 1; es decir, el valor de la variable de control se irá aumentando de 1 en 1 por cada ejecución de las instrucciones que conformen el bucle.
La instrucción FOR es la primera instrucción de cualquier bucle.
El formato de la instrucción NEXT es el siguiente:



La instrucción NEXT es la que cierra siempre un bucle, por lo tanto, es su última instrucción y tiene relación con la instrucción FOR que es la primera.

NEXT — Nombre de la instrucción.

Variable de control — Nombre de la variable de control especificada en la primera instrucción del bucle. Cada vez que se pasa por la instrucción NEXT, se aumenta el valor de esta variable en un valor igual al especificado en el parámetro «incremento» de la instrucción FOR. Este parámetro es variable, ya que con la palabra NEXT es suficiente. El BASIC asocia el primer NEXT que encuentra con el último FOR que ha encontrado.

Ejemplos

```
30 FOR I=1 TO 5
40 PRINT "PEPE"
50 NEXT I
60 END
```

Este pequeño programa visualizaría 5 veces la palabra PEPE en la pantalla. El valor de la variable I empezará siendo 1; después de visualizar el mensaje PEPE y al ejecutarse la instrucción NEXT, será 2 y así sucesivamente hasta que I sea igual a 6, con lo que terminará el bucle.
Veamos ahora cómo podría constituirse este mismo programa, pero sin utilizar las instrucciones FOR/NEXT para construir el bucle.

```
30 I=1
40 IFI >5 THEN GOTO 80
50 PRINT "PEPE"
60 I=I+1
70 GOTO 40
80 END
```

Notemos que en este caso hemos escrito más instrucciones, la 30, 40, 60 y 70, cuando en el caso anterior solamente con la 30 y 50 había suficiente.

```
30 FOR I=1 TO 5 STEP 2
40 PRINT "PEPE"
50 NEXT I
60 END
```

En este caso la palabra PEPE sólo se visualizará tres veces, ya que el incremento de la variable de control (I) no será de 1 en 1, sino de 2 en 2.
En la 1.ª pasada por el bucle valdrá 1, en la segunda valdrá 3, en la tercera valdrá 5 y en la cuarta 7, con lo que será mayor de 5 y por lo tanto el bucle habrá terminado.
Veamos cómo sería el mismo programa pero sin utilizar las instrucciones FOR/NEXT:

```
30 I=1
40 IFI > 5 THEN GOTO 80
50 PRINT "PEPE"
60 I=I+2
70 GOTO 40
80 END
```

Como podemos ver, solamente cambia con respecto a su homónimo anterior, la instrucción 60, que en el primer caso es I=I+1 y en éste es I=I+2.

Las instrucciones NEXT de los ejemplos vistos, línea 50 NEXT I, pueden sustituirse por NEXT sin la variable.

```
30 FOR I=1 TO 5
40 PRINT "PEPE"
50 NEXT
60 END
30 FOR I=1 TO 5 STEP 2
40 PRINT "PEPE"
50 NEXT
60 END
```

Cuando en un programa BASIC se ejecuta una instrucción NEXT, sin que antes se haya ejecutado una FOR, se producirá un error.
Por ejemplo, en este caso:

```
30 GOTO 80
40 FOR X=1 TO 80
50 PRINT "JOSE"
80 NEXT
90 END
```

Vemos aquí que la primera instrucción en la línea 30 cede el control a la 80 y en la 80 hay un NEXT, por lo tanto, se ejecutará una instrucción NEXT sin haberse ejecutado antes la FOR y esto producirá un error.

FUNCIONES DEL BASIC

En el BASIC existen dos tipos de funciones: las propias del lenguaje BASIC y las que puede definir el propio usuario.
En este apartado nos ocuparemos fundamentalmente de las del propio BASIC. Las de usuario, al poder éste definirlas, puede darles el nombre que quiera; por lo tanto no serán fijas, sino que dependerá del propio usuario.
Las *funciones de usuario* podríamos decir que son como instrucciones que puede definir el propio usuario; a partir del mismo momento que las define, puede utilizarlas ya como si fueran instrucciones. Por ejemplo si escribimos la línea 30 DEF FN(X)=(X/5)—4, a partir de aquí podemos escribir, por ejemplo, otra línea de programa como ésta 60 A= FN(40); cuando se ejecute esta línea, es como si se estuviera ejecutando ésta 60 A= (40/5)—4.

FUNCIONES PROPIAS DEL BASIC

Las funciones propias del lenguaje BASIC representan una parte muy importante, tanto en cantidad como en calidad, de tal lenguaje. Mediante estas funciones pueden efectuarse operaciones de tipo numérico o alfanumérico.
No las describiremos todas, sino que explicaremos su comportamiento y sólo describiremos a fondo las más importantes.
Tales funciones se dividen en tres tipos:
— numéricas;
— alfanuméricas;
— de entrada y salida especiales.

Funciones numéricas

ABS — Calcula el valor absoluto de una expresión numérica.
Ejemplo 10 A=ABS(7*(-2))

COS — Calcula el coseno del argumento.
Ejemplo 10 A=COS (0.5)

FIX — Calcula la parte entera de una expresión numérica.
Ejemplo 10 PRINT FIX(60.78), visualizará 60.

FRE — Calcula el espacio de memoria que queda libre después de haber cargado el programa.
Ejemplo 10 A=FRE(Ø)

INT — Calcula el entero más grande, inferior o igual a la expresión numérica especificada.
Ejemplo 10 PRINT INT (65.75); visualizará 65

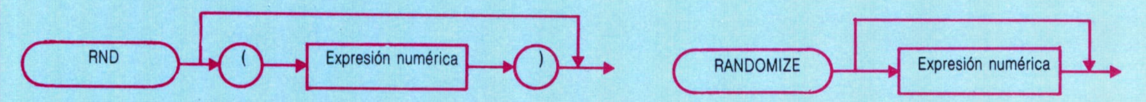
LOG — Calcula el logaritmo natural de una expresión numérica positiva.
Ejemplo 10 A=LOG(40/6)

SGN — Retoma un 1 si la expresión numérica es positiva y un Ø si es negativa.
Ejemplo 10 A=SGN(-5); en A se almacenará un Ø, porque el signo de la expresión es negativo.

SIN — Calcula el seno de una expresión numérica.
Ejemplo 10 A=SIN(0.5)
SQR — Calcula la raíz cuadrada de una expresión numérica.
Ejemplo 10 A=SQR(25)
TAN — Calcula la tangente de una expresión numérica.
Ejemplo 10 A=TAN(2/5)

Funciones RND y RANDOMIZE

Retornan un número casual comprendido entre Ø y 1. Si se ejecuta un programa y dentro de este programa se pasa 10 veces por una instrucción RND, se generará una secuencia de 10 números entre Ø y 1. Si se vuelve a ejecutar el mismo programa, la secuencia de los 10 números será la misma, es decir, que se generarán los mismos 10 números. Si queremos que esto no ocurra, se utilizará la función RANDOMIZE.
Si especificamos la función RANDOMIZE al principio del programa, ésta hará que cambie la secuencia cada vez que se ejecuta el programa.
Sus formatos son:



RND — Nombre de la función.
Expresión numérica — Si es < Ø, comienza otra vez la misma secuencia de números casuales. Si es =Ø, repite el último número generado. Si es > Ø, se genera el siguiente número casual. Este caso es el mismo que si no se especifica la expresión numérica.

RANDOMIZE — Nombre de la función.
Expresión numérica — Se utiliza para definir el punto inicial de una nueva secuencia de números casuales.

Ejemplos

Supongamos el siguiente programa:

```
10 FOR I=1 TO 4
20 PRINT RND;
30 NEXT
40 END
```

Al ejecutarlo tres veces, en la pantalla se visualizará lo siguiente:

```
0.885051    0.252544   0.461258   0.994602
0.885051    0.252544   0.461258   0.994602
0.885051    0.252544   0.461258   0.994602
```

Las tres series de números generadas son las mismas. Sin embargo, si en este mismo programa especificamos la función RANDOMIZE, la secuencia cambia.

```
10 INPUT A
20 RANDOMIZE A
30 FOR I=1 TO 4
40 PRINT RND;
50 NEXT
60 END
```

Al ejecutar este programa, en la primera instrucción se espera que se entre desde teclado una cantidad numérica que será asignada a la variable A; dicha cantidad o valor servirá para dar el cambio de secuencia a la instrucción RANDOMIZE. De esta manera, después de ejecutar tres veces este programa, esperará la entrada de la cantidad. Si le damos sucesivamente los valores 3, 4 y 5, podríamos obtener el siguiente resultado:

```
0.121456   0.252412   0.949518   0.333215
0.141619   0.987645   0.557619   0.212112
0.331455   0.786101   0.221479   0.441678
```

Funciones alfanuméricas

ASC — Calcula un valor numérico que es el código decimal ASCII del primer carácter de una expresión alfanumérica.
Ejemplo 10 X$="TEXTO"
20 PRINT ASC(XS)

Al ejecutar estas dos instrucciones, en la pantalla se visualizará el valor 84.

CHR$ — Transforma un código decimal ASCII en su correspondiente caracter alfanumérico.
Ejemplo 10 PRINT CHR$(84); dará como resultado T.

LEN — Calcula la longitud en número de caracteres de una serie alfanumérica dada.
Ejemplo 10 A$="MARC"
20 A=LEN(A$). En este caso, A tomará el valor 4.

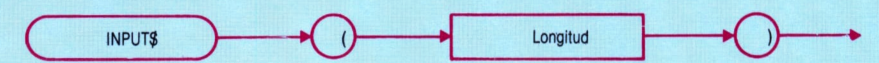
SPACE$ — Crea una serie alfanumérica de una longitud específica y la rellena con caracteres blancos:
Ejemplo 10 A$=SPACE$(10)

STR$ — Convierte el valor de una expresión numérica en una serie alfanumérica.
Ejemplo 10 A$=STR$(1560)
20 PRINT A$. En este caso A$ será igual a "1.560".

VAL — Convierte una serie alfanumérica en numérica.
Ejemplo 10 A$="1560"
20 A=VAL(A$). En este caso, A será igual a 1.560

La función INPUT$

Mediante esta función pueden entrarse datos alfanuméricos desde teclado.
Su formato es el siguiente:



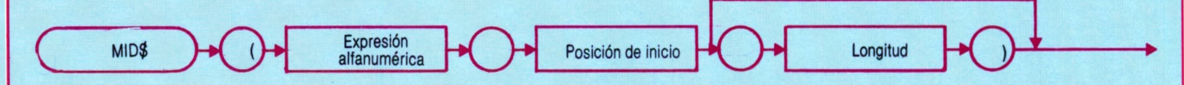
INPUT$ — Nombre de la función.
Longitud — Número de caracteres alfanuméricos que se pueden entrar desde teclado. Su valor no puede ser superior a 255.

Ejemplo

10 A$=INPUT$ (30)
Desde teclado se digitarán treinta caracteres y se almacenarán en la variable alfanumérica A$.

La función MID$

Realiza la extracción de una subserie dentro de una serie de caracteres alfanuméricos, empezando desde una posición determinada.
Su formato es el siguiente:



MID$ — Nombre de la función.
Expresión alfanumérica — Puede ser el nombre de una variable alfanumérica o una constante alfanumérica de la que se quiere extraer una parte.
Posición de inicio — Valor de la posición que ocupa el primer carácter a extraer, empezando por la izquierda de la serie alfanumérica.
Longitud — Número de caracteres que se quieren extraer. Este parámetro es opcional; si no se especifica, se extraerán todos los caracteres desde el de la posición de inicio hasta el final de la serie alfanumérica.

Ejemplos de todos los casos

10 A$="PEPITA JIMÉNEZ"
20 B$= MID$(A$, 8)

Ejecutando estas dos instrucciones, extraeremos de la serie alfanumérica almacenada en A$ una subserie, empezando desde la posición 8 hasta el final, porque no hemos especificado el parámetro longitud. Entonces, en B$ se almacenará la subserie "JIMÉNEZ".

10 A$= "PEPITA JIMÉNEZ"
20 B$= MID$(A$, 1,6)

Sin embargo, en este caso el valor de B$ será la subserie "PEPITA", porque hemos especificado, desde la posición 1, 6 caracteres.

Funciones de entrada/salida y especiales

Estas funciones facilitan la ejecución de las operaciones de entrada/salida, las conversiones de valores, la gestión de los errores, etcétera. Se describen a continuación.

CVD — Convierte una serie de caracteres en un número en doble precisión. Se utiliza para

convertir campos de 8 bytes de un récord, leído de un fichero directo, a números de doble precisión.
Ejemplo 10 A£=CVD(A$). Siendo A$ un campo de 8 bytes de un récord.

CVS — Convierte una serie de caracteres en un número de simple precisión. Se utiliza para convertir campos de 4 bytes de un récord, leído de un fichero directo, a números de simple precisión.
Ejemplo 10 A!=CVS(B$). Siendo B$ un campo de 4 bytes de un récord.

CVI — Convierte una serie de caracteres en un número entero. Se utiliza para convertir campos de 2 bytes de un récord, leído de un fichero directo, a números enteros.
Ejemplo 10 A%=CVI(C$). Siendo C$ un campo de 2 bytes de un récord.

LPOS — Devuelve la posición actual de la cabeza de impresión de la impresora.
Ejemplo 10 IF LPOS(Y) > 132 THEN LPRINT

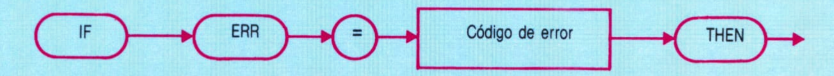
MKD$ — Convierte un número de doble precisión en una serie de 8 caracteres. Es la función inversa de la función CVS y se aplica cuando se quiere almacenar en un récord un número de doble precisión.
Ejemplo 10 D$=MKD$(A£)

MKS$ — Convierte un número de simple precisión en una serie de 4 caracteres. Es la función inversa de la función CVS y se aplica cuando se quiere almacenar en un récord un número de simple precisión.
Ejemplo 10 A$=MKS$(A!)

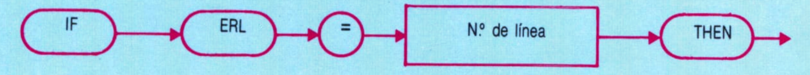
MKI$ — Convierte un número entero en una serie de 2 caracteres. Es la función inversa de la función CVI y se aplica cuando se quiere almacenar en un récord un número entero.
Ejemplo 10 B$=MKI$(A%)

LAS FUNCIONES ERR, ERL Y LAS INSTRUCCIONES ERROR GOTO Y RESUME

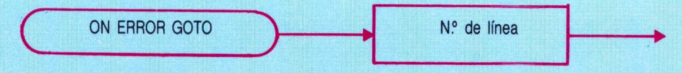
Estas dos funciones y estas dos instrucciones siempre se encuentran juntas en un programa BASIC y sirven para efectuar el tratamiento de los errores que puedan producirse en tales programas.
La función ERR devuelve un código de error cuando se produce alguna anomalía en la ejecución de un programa BASIC. Además de función, hace de variable, ya que es en ella misma donde se almacena el código de error.
Cuando se produce un error en la ejecución de un programa BASIC, la función ERL devuelve el número de línea en donde se ha producido el error. Esta función se comporta también como la función ERR, ya que, a su vez, es la variable donde se almacena el número de línea.
La instrucción ON ERROR GOTO siempre se especifica en las primeras líneas de programa y es una instrucción que solamente se ejecuta cuando se produce un error de ejecución en el programa que la contiene. Es decir, cuando se produce un error, se le da el control a la línea especificada después de la palabra GOTO y a partir de esa línea se hace el tratamiento de tal error.
La instrucción RESUME sirve para dar el control de ejecución a una instrucción determinada, después de haber efectuado el tratamiento de tal error.
Sus formatos son los siguientes:



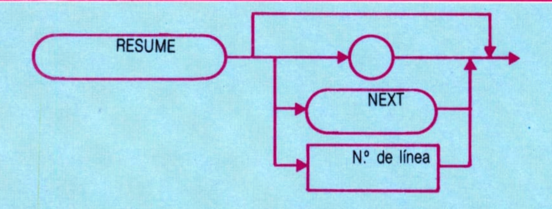
En la verificación de errores, la función ERL también forma parte siempre de una instrucción IF.
ERL — Nombre de la función.
Nº de línea — Número de línea donde puede haberse producido el error.



En la verificación de errores, la función ERR siempre formará parte de una instrucción IF.
ERR — Nombre de la función.
Código de error — Número del error esperado.



ON ERROR GOTO — Nombre de la instrucción.
Nº de línea — Número de línea del programa a la que se le cede el control automáticamente cuando se produzca un error de ejecución en cualquier parte del programa. Normalmente, a partir de esta línea de programa se encontrará la parte de programa que realiza el tratamiento de los errores.



RESUME — Nombre de la instrucción.
0 — Se cede el control a la línea donde se ha producido el error.
NEXT — Se cede el control a la línea siguiente a la que se ha producido el error.
N.º de línea — Se cede el control al número de línea especificado.

Ejemplos

Como estas funciones e instrucciones se complementan, vamos a describir un ejemplo de un programa donde se podrá ver el uso de tales funciones y ediciones para el tratamiento de los errores que se pueden producir en este programa.
Un error común de cálculo es la división por cero dentro de un programa BASIC. Veamos entonces cómo podemos tratar ese error. Para ello, supongamos que el código de tal error es el 10.

```
 10 ON ERROR GOTO 200
 20 INPUT A,B,C
 30 D=B/(C-A)
 40 PRINT D
 50 A=A+1
 60 IF A > 200 THEN GOTO 500
 70 GOTO 30
200 IF(ERR=10) AND (ERL=30) THEN A=A+1: RESUME Ø
210 PRINT "ERROR N.º ";ERR;" EN LÍNEA"; ERL
500 END
```

En la instrucción 10 especificamos que si se produce cualquier error en el programa, el control se cede a la línea 200; a partir de esta línea se efectúa el tratamiento de errores. En la 200 se investiga si el error es el n.º 10 (división por cero) y si se ha producido en la línea 30. Si es así, sumamos uno al valor de la variable A y cedemos el control mediante la sentencia RESUME 0 a la misma línea de programa donde se ha producido el error, para que se vuelva a ejecutar la operación; al haberle sumado 1 a la variable A, ya no volverá a producirse el error anterior.
Si el error que se ha producido no ha sido el n.º 5 o no se ha producido en la línea 30, entonces, mediante la instrucción 210, se visualizará en la pantalla el n.º de error y la línea donde se ha producido, con este formato: ERROR N.º ______ EN LÍNEA ______. Después finalizará la ejecución del programa con la línea 500.
Si en lugar de RESUME 0 hubiéramos especificado RESUME NEXT, caso de producirse el error en la línea 30, se ejecutarían las instrucciones de la línea 200 y se devolvería el control a la línea siguiente a la 30; pero, en este caso, este proceder no tiene mucho sentido.
En cambio, si hubiésemos construido la línea 200 de esta manera:

```
200 IF(ERR=10) AND (ERL=30) THEN RESUME 50
```

casi sería el mismo tratamiento que con RESUME 0, ya que en este caso no sumamos un 1 a la variable A, sino que cedemos el control a una instrucción que hace lo mismo.

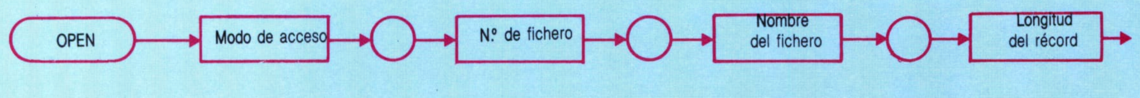
INSTRUCCIONES DE MANEJO DE FICHEROS EN DISCO

El lenguaje BASIC permite manejar dos tipos de ficheros o archivos:
— ficheros de acceso secuencial;
— ficheros de acceso directo.

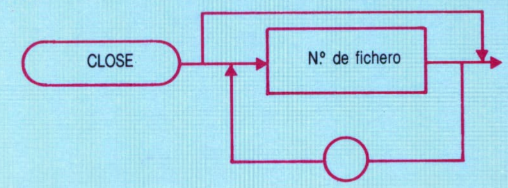
En BASIC, los ficheros secuenciales normalmente son ficheros de texto (cartas, memorándums, etc.) y no vamos a ocuparnos de las instrucciones que los manejan, porque nos parecen menos importantes que las que soportan los ficheros de acceso directo, aunque algunas de ellas son las mismas. En este caso sí diremos que sirven tanto para unos como para otros.
Los ficheros directos que pueden manejarse en el BASIC se describen en «Ficheros de organización directa» del apartado «Organización de los datos» del capítulo «Conceptos básicos en la programación de computadoras».
Veamos las instrucciones más importantes para el manejo de estos ficheros.

Las instrucciones OPEN y CLOSE

Cualquier fichero, en cualquier lenguaje, se tiene que abrir para poder tener acceso a él; de la misma manera, cuando ya no se quiera usar más, se tiene que cerrar. Las instrucciones del BASIC que realizan estas funciones son las instrucciones OPEN (abrir) y CLOSE (cerrar). Estas dos funciones operan tanto en ficheros directos como secuenciales.
Su formato es el siguiente:



OPEN — Nombre de la instrucción.
Modo de acceso — Puede especificarse Secuencial o Random (Directo)
R para Random o Directo
A, E, I, O para Secuencial
N.º de fichero — Número que se asigna a cada fichero que se abre. Este número no puede ser el mismo dado a otro fichero que se haya abierto en el mismo programa.
Nombre del fichero — Nombre del fichero que se quiere abrir.
Longitud del récord — Longitud en caracteres o bytes del récord lógico del fichero que se quiere abrir.



CLOSE — Nombre de la instrucción.
N.º de fichero — Número que se le ha dado al fichero que se quiere cerrar cuando se ha abierto. Si este parámetro se omite, entonces se cerrarán todos los ficheros que estén abiertos.
Para ver un ejemplo claro, supongamos que hemos abierto cuatro ficheros con los números 1, 2, 3 y 4. Podemos cerrarlos todos de dos maneras:
CLOSE o bien CLOSE 1, 2, 3, 4

Ejemplos

```
 10 OPEN "R", 1, "PEPE", 128
  :
100 CLOSE
```

En este primer caso, abrimos el fichero PEPE de acceso directo ("R") con número 1 y con longitud de récord de 128 bytes; en la instrucción 100 lo cerramos.

```
 10 OPEN "R", 1, "PEPE", 128
 20 OPEN "R", 2, "PIPO", 256
  :
100 CLOSE 1
110 CLOSE 2
```

En el segundo caso, abrimos el mismo fichero anterior y otro llamado PIPO con el número 2; en la instrucción 100 cerramos el primero y en la 110 el segundo. Sin embargo, estas dos últimas instrucciones las podríamos sustituir por una sola 100 CLOSE, que cerraría los dos ficheros a la vez.

Las instrucciones GET/PUT

Las instrucciones GET y PUT se utilizan para leer y grabar respectivamente récords lógicos en un fichero de acceso directo.
Su formato es el siguiente:



GET — Nombre de la instrucción.
N.º de fichero — Número que se ha asignado al fichero que se quiere leer, cuando se ha abierto.

Nº de récord — Número de récord lógico que se quiere leer.
PUT — Nombre de la instrucción.
Nº de fichero — Número que se ha asignado al fichero que se quiere grabar, cuando se ha abierto.
Nº de récord — Número de récord lógico que se quiere grabar.

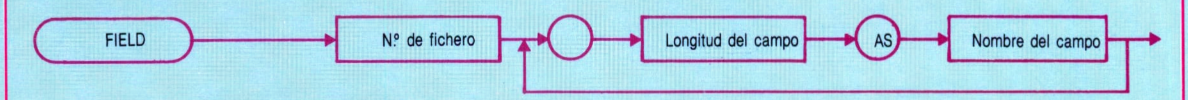
Ejemplos

```
 10 OPEN "R", 1, "PEPE", 128
  :
100 GET 1,5
  :
200 PUT 1,5
  :
300 CLOSE
```

En este esquema de programa vemos que en la línea 10 abrimos el fichero PEPE y le asignamos el número 1. Como ya hemos explicado anteriormente, esta instrucción es obligatoria para poder más tarde leer y grabar récords de un fichero. En la línea 100 efectuamos la lectura del récord número 5 y en la línea 200 grabamos el mismo récord; finalmente, en la línea 300 cerramos el fichero, instrucción también obligatoria.
Sin embargo, observando este programa pueden surgir un par de dudas: cuando se ejecuta una instrucción GET y se lee un récord, dónde colocar la información de este récord para que la podamos consultar o cambiar y, al revés, para grabar un récord, dónde tener la información de este récord para que se grabe en el fichero.
Estas dudas quedarán despejadas a continuación, porque vamos a describir la instrucción FIELD y sus asociadas, que precisamente tienen que ver con ellas.

La instrucción FIELD

La instrucción FIELD sirve para definir una zona de memoria o área de trabajo que se corresponderá con la longitud de un récord de un fichero y en la que se almacenarán los records, al leerlos o antes de grabarlos en tal fichero.
Habrá que definir un área para cada fichero; por lo tanto, en un programa en el que se manejan 5 ficheros de acceso directo, se especificarán 5 instrucciones FIELD.
Esta área de trabajo puede estar formada por un solo campo o por varios campos y es completamente alfanumérica.



FIELD — Nombre de la instrucción.
Nº de fichero — Número con el que se ha abierto el fichero del cual estamos definiendo su área de trabajo.
Longitud del campo — Longitud, en bytes o caracteres, de cada uno de los campos de que puede constar el área de trabajo.
Nombre del campo — Nombre que servirá para identificar cada uno de los campos del área de trabajo. Siempre serán nombres de variables alfanuméricas.

Ejemplos

```
20 FIELD 1, 128 AS A$
```

En la primera instrucción estamos definiendo el área de trabajo del fichero que se ha abierto con el número 1, cuya longitud (que se corresponderá con la longitud del récord del fichero) es de 128. Como vemos, estamos definiendo un solo campo AS de longitud 128.

```
20 FIELD 1, 8 AS A$, 30 AS B$, 30 AS C$, 30 AS D$, 8 AS E$, 8 AS F$, 8 AS G$, 4 AS H$, 2 AS I$
```

En esta otra instrucción estamos definiendo el área de trabajo para el mismo fichero, de la misma longitud; pero en este caso, en lugar de definir un solo campo, hemos definido 9 campos, cada uno de ellos con su longitud, de tal manera que la suma de las longitudes de todos ellos es igual a la longitud del área de trabajo, es decir 128.

A$ → 8 bytes	D$ → 30 bytes	G$ → 8 bytes
B$ → 30 bytes	E$ → 8 bytes	H$ → 4 bytes
C$ → 30 bytes	F$ → 8 bytes	I$ → 2 bytes

El acceso a la información contenida en cada uno de los campos una vez leído un récord, o el almacenamiento de la información en estos campos antes de grabar un récord, se realiza a través de las instrucciones y funciones siguientes, cuyo uso ya se ha descrito anteriormente.

Después de ejecutar una instrucción GET, la información del registro leído se almacenará en el área de trabajo. En esta área de trabajo todos los campos serán alfanuméricos; sin embargo, habrá informaciones numéricas convertidas, de manera que con las funciones CVI, CVS y CVD podremos obtener su valor numérico.

- Un valor entero ocupará 2 bytes.
- Un valor de simple precisión ocupará 4 bytes.
- Un valor de doble precisión ocupará 8 bytes.

Supongamos que en el ejemplo anterior, en el que definíamos una instrucción FIELD con 9 campos, los tres últimos campos, de 8, 4 y 2 bytes respectivamente, podrían tener almacenados: el primero, G$, un valor de doble precisión; el segundo, H$, un valor de simple precisión, y el tercero, I$, un valor entero.
Para obtener estos valores numéricos en variables numéricas haríamos lo siguiente:

```
10 A£=CVD(G$): A!=CVS(H$): A%=CVI(I$)
```

Con lo que nos quedaría en A£, A! y A% el valor real de estos tres campos.
En lo que respecta a los campos realmente alfanuméricos, se pueden obtener, tal como se encuentran en el área de trabajo, simplemente haciendo referencia a su nombre.
En cuanto a cómo se deben almacenar los campos de esta área de trabajo para luego grabar el récord, podemos decir lo siguiente: para almacenar cualquier campo se usarán siempre las instrucciones LSET o RSET, según queramos que el campo quede alineado por la izquierda o por la derecha.
Para almacenar un valor alfanumérico es suficiente con la instrucción LSET o RSET.
Ejemplo: si en B$ tenemos un nombre y lo queremos almacenar en el campo A$ (nombre) del área de trabajo, para posteriormente realizar una grabación, escribiremos

```
10 LSET A$=B$
```

Hemos especificado LSET porque queremos que este valor quede alineado por la izquierda.
Sin embargo, para almacenar un valor numérico entero, utilizaremos la función MKI$, además de LSET o RSET.
Ejemplo: supongamos que en A% tenemos un valor y lo queremos almacenar en el campo I$ del área de trabajo, entonces escribiremos

```
20 LSET I$=MKI$(A%)
```

Igual ocurrirá con valores numéricos de simple y doble precisión, pero con instrucciones diferentes: MKS$ y MKD$

```
30 LSET H$=MKS$(A!)
40 LSET G$=MKD$(A£)
```

Pensemos siempre que, al hacer esto, forzosamente I$ debe ser un campo de 2 bytes de longitud, H$ de 4 y G$ de 8.
A continuación veremos un ejemplo concreto del tratamiento completo de un fichero, para que podamos tener una idea global de cómo se hace.
Vamos a leer 100 registros de un fichero llamado «ARTIC», en el que cada récord tiene almacenados datos de un artículo de orfebrería diferente, y le vamos a modificar el campo *precio* subiéndolo un 20 %.
Los campos del récord serán los siguientes:

AR$ — Código de artículo de 8 bytes.
DE$ — Descripción de 30 bytes.
PR$ — Precio de 4 bytes.
EX$ — Existencia de 2 bytes.

En total, la longitud del récord será de 44 bytes.
He aquí el programa para realizar este proceso:

```
 10 REM EJEMPLO DEL TRATAMIENTO DE UN FICHERO DIRECTO
 20 OPEN "R",1,"ARTIC", 44        Apertura del archivo.
 30 FIELD 1,8 AS AR$, 30 AS DE$, 4 AS PR$, 2 AS EX$        Definición del área de trabajo del fichero.
 40 FOR X=1 TO 100        Inicio de bucle para ir leyendo del récord 1 al 100.
 50 GET 1,X        Lectura del récord iniciado por X.
 60 A!=CVS(PR$)        Obtención del precio en numérico desde el campo PR$ del área de trabajo.
 70 B!=(A!*20)/100        Cálculo del nuevo precio.
 80 A!=A!+B!
 90 LSET PRS=MKS$(A!)        Almacenamiento del nuevo precio (numérico de simple precisión)
                             en el campo PRS del área de trabajo.
100 PUT 1,X        Grabación del mismo récord que se ha leído, pero con el nuevo precio.
110 NEXT        Última instrucción del bucle; hace incrementar el valor de la variable X hasta
                que sea mayor de 100, caso en el que habría finalizado el bucle.
120 CLOSE        Cierre del fichero una vez finalizado el proceso.
130 END        Fin del programa.
```

INSTRUCCIONES GRÁFICAS

Desde hace pocos años los lenguajes BASIC van evolucionando a medida que evolucionan las computadoras. Debido a que cada día van apareciendo más computadoras que poseen pantallas con posibilidades gráficas, surge la necesidad de que los lenguajes que pueden soportar estas computadoras tengan posibilidades de manejar este tipo de pantallas y puedan realizar, mediante las instrucciones correspondientes, dibujos y gráficos.
No vamos a describir esta parte del BASIC, porque pensamos que es mejor, primero, aprender las instrucciones fundamentales y, después, aprender a manejar este grupo que es complementario. Lo que sí haremos será citar algunas de ellas con la función que realizan:

LINE — Traza una línea recta o dibuja un rectángulo, según los parámetros que se le den.
Ejemplo: LINE (100,200)—(200,200) → traza una línea
LINE (100,200)—(200,250),,B → dibuja un rectángulo

CIRCLE — Traza una circunferencia de un radio especificado.
Ejemplo: CIRCLE (100,200),70 → dibuja un círculo de radio 70.

DRAW — Es la instrucción gráfica más potente y con ella se puede hacer cualquier tipo de dibujo. Supongamos, en vídeo, una hoja de papel y que estamos haciendo dibujos con el lápiz. Pues bien, la instrucción DRAW se comporta igual con la pantalla.
Ejemplo: DRAW «M=X=,25»
Si suponemos que el lápiz se encuentra en el punto (10, 20), esta instrucción hace que se desplace, trazando una línea, a la posición (33,45), es decir (23+10,20+25).

PAINT — Colorea el interior de una figura (un rectángulo, una circunferencia, etc.) con el color que se le especifique.
Ejemplo: Si hemos dibujado la circunferencia con la instrucción
10 CIRCLE (100,200),70
Con ella coloreamos tal circunferencia
20 PAINT (110,180)
es decir, especificamos un punto interior de la circunferencia.

ALGUNOS EJEMPLOS DE PROGRAMAS ESCRITOS EN LENGUAJE BASIC

Para mostrar unos cuantos ejemplos de programas BASIC, utilizaremos algunos de los diagramas de flujo construidos en el capítulo «Organigramas de algoritmos numéricos». De esta forma, veremos el análisis del problema, su resolución mediante un diagrama y su transformación de diagrama a programa BASIC.
En todos los casos, primero estudiaremos cómo se ha resuelto el problema con el diagrama y, una vez comprendido esto, veremos cómo se ha transformado en programa BASIC.

Ejemplo 1

Veamos cómo transformamos a programa BASIC el problema resuelto con los diagramas de las **figuras 11** y **12** (véase pp. 189 y 190). Se trata de calcular cambio de moneda extranjera. Fijémonos que de este problema se han construido dos diagramas, el de la **figura 11** y el de la **12**.

Programa correspondiente al diagrama de la **figura 11** *(pág. 189)*.

```
10 INPUT "CAMBIO LIRA";I
20 INPUT "CAMBIO FRANCO";F
30 INPUT "CAMBIO DOLAR";D
40 INPUT "CAMBIO LIBRA";L
50 INPUT "TIPO DE CAMBIO";A
60 INPUT "CANTIDAD";B
70 IF A=1 THEN GOTO 130
80 IF A=2 THEN GOTO 150
90 IF A=3 THEN GOTO 170
100 IF A=4 THEN GOTO 190
110 PRINT "ERROR EN TIPO DE CAMBIO"
120 GOTO 50
130 C=B*D
140 GOTO 200
150 C=B*L
160 GOTO 200
170 C=B*F
180 GOTO 200
190 C=B*I
200 PRINT C; "U.M."
```

```
210 INPUT "MAS CAMBIO";Z
220 IF Z <> 0 THEN GOTO 50
230 END
```

Programa correspondiente al diagrama de la **figura 12** *(pág. 190).*

```
 10 INPUT "CAMBIO LIRA";I
 20 INPUT "CAMBIO FRANCO";F
 30 INPUT "CAMBIO DOLAR";D
 40 INPUT "CAMBIO LIBRA";L
 50 INPUT "TIPO DE CAMBIO";A
 60 INPUT "CANTIDAD";B
 70 ON A GOTO 100,120,140,160
 80 PRINT "ERROR EN TIPO DE CAMBIO"
 90 GOTO 50
100 C=B*D
110 GOTO 170
120 C=B*L
130 GOTO 170
140 C=B*F
150 GOTO 170
160 C=B*I
170 PRINT C; "U.M."
180 INPUT "MAS CAMBIO»;Z
190 IF Z <> Ø THEN GOTO 50
200 END
```

Antes que nada, señalemos que la pregunta ¿más cambio?» del diagrama aquí la hemos sustituido por una entrada, de manera que si el usuario entra un 0, significa que no quiere hacer más cálculos y termina el programa; por el contrario, si digita otro valor, se vuelve a iniciar el proceso.
La única diferencia entre estos dos programas es la misma que en los diagramas: las condiciones. Mientras en el primero se especifican cuatro instrucciones de la línea 70 a la 100, en el segundo con una conseguimos lo mismo en la línea 70.

Ejemplo 2

El siguiente programa BASIC corresponde a la **figura 24** *(pág. 194)*; se trata de introducir unos datos (edades de alumnos) en un vector de 10 elementos y, una vez introducidos, dar como resultado la media aritmética de todos los elementos.
En este caso también escribiremos dos programas: el primero, siguiendo el diagrama y el segundo, utilizando instrucciones más complejas para reducir tiempo.

1.er programa

```
  5 DIM IN(10)
 10 X=1
 20 INPUT IN(X)
 30 X=X+1
 40 IF X > 10 THEN GOTO 60
 50 GOTO 20
```

Entrada de las edades y asignación de éstas en un vector

```
 60 A=0
 70 I=1
 80 A=A+IN(I)
 90 I=I+1
100 IF I > 10 THEN GOTO 120
110 GOTO 80
```

Suma del valor de todos los elementos del vector.

```
120 B=A/10
```

Cálculo de la media.

```
130 PRINT B
140 END
```

Visualización del resultado y fin del programa.

2.º programa

```
 5 DIM IN(10)
10 FOR X=1 TO 10
20 INPUT IN(X)
30 NEXT
```

Entrada de las edades y asignación de éstas en un vector.

```
40 FOR I=1 TO 10
50 A=A+IN(I)
60 NEXT
```

Suma del valor de todos los elementos del vector.

```
70 B=A/10
```

Cálculo de la media

```
80 PRINT B
90 END
```

Visualización del resultado y fin del programa.

Si comparamos los dos programas, se puede ver la utilidad de las instrucciones FOR/NEXT, con las que se gana espacio y tiempo en la escritura de programas.

Ejemplo 3

Veremos ahora el mismo ejemplo anterior, pero en este caso las edades las tendremos en una matriz.

```
  5 DIM A(5,2)
 10 FOR X=1 TO 2
 20 FOR Y=1 TO 5
 30 INPUT A(Y,X)
 40 NEXT
 50 NEXT
```

Entrada de las edades y asignación de éstas en una matriz.

```
 60 FOR CLAS=1 TO 2
 70 FOR NUM=1 TO 5
 80 B=B+A(NUM,CLAS)
 90 NEXT
100 NEXT
```

Suma del valor de todos los elementos de la matriz.

```
110 C=B/10
```

Cálculo de la media.

```
120 PRINT C
130 END
```

Visualización del resultado y fin del programa.

Ejemplo 4

Veamos un ejemplo de tratamiento de ficheros directos; en él apreciamos cómo se manejan tales ficheros y sus récords lógicos y también cómo se trabaja con subseries de series alfanuméricas. Supongamos que tenemos un fichero directo con 2.000 récords, en el que en cada uno de esos récords se encuentran almacenados datos relativos a cada uno de los socios de una entidad, con este formato:

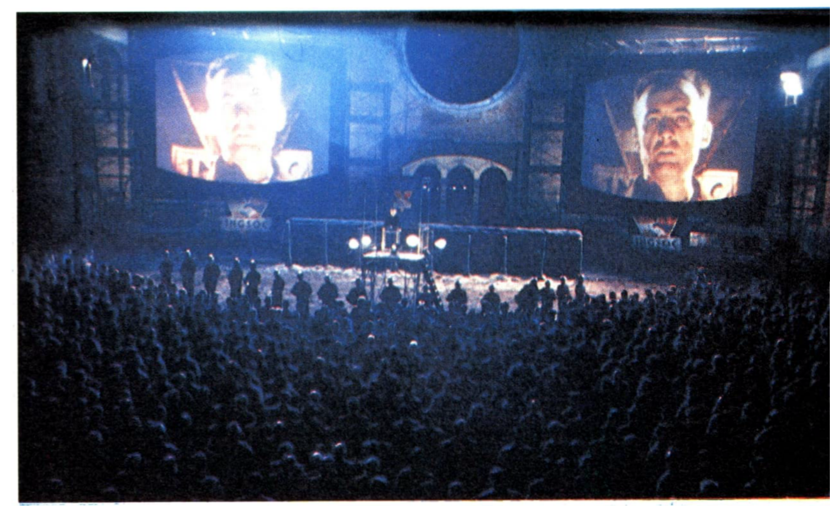
NUM$	NÚMERO DE SOCIO	4 bytes	
NOM$	NOMBRE Y APELLIDOS	30 bytes	
DIR$	DIRECCIÓN	30 bytes	
POB$	POBLACIÓN	29 bytes	97 bytes
TEL$	TELÉFONO	10 bytes	
TIP$	TIPO DE SOCIO Y CUOTA	3 bytes	

Este último campo necesita una explicación. Consta de 3 bytes. El primero de ellos puede tener dos valores: V de varón o H de hembra. El segundo puede tener también dos valores: J de jubilado y N de no jubilado. El tercero podrá tener tres valores: A de pago anual, T de pago trimestral y M de pago mensual.
Una vez descrita la composición del récord lógico de este fichero, pasemos a exponer el problema que queremos solucionar.
Queremos obtener, de este fichero, un listado de todos los socios jubilados para saber cuántos hay en la entidad y enviarles un pequeño obsequio.

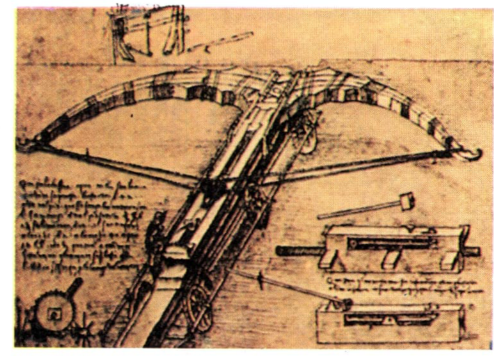
```
10 OPEN "R",1, "SOCIOS",97
20 FIELD 1,4 AS NUM$, 30 AS NOM$, 30 AS DIR$, 20 AS POB$, 10 AS TEL$, 3 AS TIP$
30 FOR N=1 TO 2000
40 GET 1,N
50 IF MID$(TIP$,2,1) <> "J" THEN GOTO 70
60 LPRINT NUM$;" ";NOM$;" ";DIR$;" ";POB$;" ";TEL$
70 NEXT
80 CLOSE
90 END
```

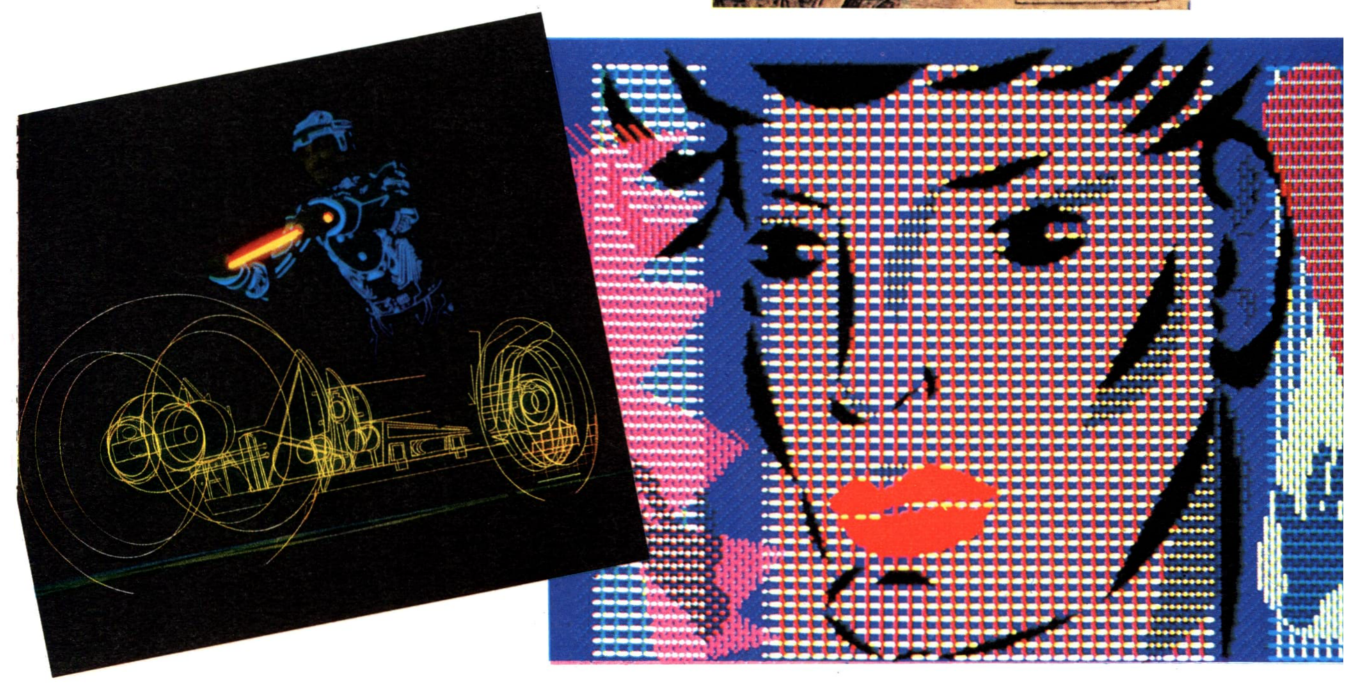
En la instrucción 10 abrimos el fichero "SOCIOS"; en la 20 definimos su área de trabajo. En la 30 construimos un bucle con la instrucción FOR, que tiene su final en la 70 con la instrucción NEXT. En la 40 leemos un récord (el que indique N). Gracias a la instrucción FOR, N irá aumentando de 1 en 1 y, por lo tanto, cada vez que ejecutemos la línea 40 leeremos el récord siguiente al leído anteriormente. En la línea 50 comprobamos si el segundo byte del campo TIP$ tiene valor "J" de jubilado; si el récord leído no tiene ese valor en ese campo, vamos a la instrucción NEXT para que aumente 1 a N y podamos leer el siguiente récord. Si, por el contrario, ese segundo byte tiene el valor "J", entonces imprimiremos por impresora una línea con todos los datos del récord y seguiremos con el bucle para leer más récords hasta que lleguemos al 2.000, que será el último leído.
Después de haber leído el 2.000, tendremos ya un listado de todos los socios jubilados con todos sus datos, que es lo que queríamos. Sólo resta cerrar el fichero y finalizar el programa, cosa que se hace en las líneas 80 y 90.





GLOSARIO





Glosario

A

ABEND *(Abnormal END)* Terminación anormal. Interrupción imprevista de algún proceso, generalmente un programa, la cual imposibilita su ejecución. (V. ABORT.)

Abort Interrupción de un programa que impide terminar su proceso normal. El control del programa se devuelve al sistema operativo. (V. ABEND.)

Abrir *(Open)* Término genérico que indica la apertura de un fichero. Tal acción debe efectuarse siempre antes de acceder a un fichero.

Acceso *(Access)* Acto por el que se accede a la información almacenada en la computadora. En los tratamientos de ficheros, el acceso puede ser *secuencial* si se llega a la información deseada después de haber recorrido todos los datos que puedan preceder a esta información, y *directo* si se accede a la información sin hacer todo el recorrido anterior, es decir, directamente.

ACIA *(Asynchronous Communications Interface Adapter)* Adaptador de interface de comunicaciones asíncronas. Placa que se coloca en la salida serial de una computadora, a la que se pueden conectar periféricos. A través de ella pueden establecerse comunicaciones asíncronas.

ACK *(Acknowledge)* Acuse de recibo. Señal de control para todas las secuencias de entrada/salida en comunicaciones, que viene representada por el carácter ASCII 6. Su envío significa normalmente la aceptación o la llegada correcta de la información recibida.

Actualización *(Updating)* Mantenimiento de una información registrada en algún dispositivo de almacenamiento. El almacenamiento o actualización de un fichero, por ejemplo, implica grabar nuevos registros, anular alguno de ellos y modificar el contenido de otros registros de tal fichero.

Acumulador *(Accumulator)* Zona o registro donde se acumulan los resultados obtenidos en las operaciones aritméticas o lógicas.

Ada Lenguaje de programación que ha tomado el nombre de Ada Augusta Lovelace Byron, hija de Lord Byron y compañera de Ch. Babbage, que incorporó numerosos conceptos al campo de la computación. Este lenguaje es concurrente, por lo que permite la multiprogramación.

AI *(Artificial Intelligence)* Inteligencia artificial.

Alfabético *(Alphabetic)* Se dice de cualquier carácter que sea únicamente una letra de la «a» a la «z». (V. TEXTO ALFABÉTICO.)

Alfabeto *(Alphabet)* En computación, se refiere al conjunto de diferentes caracteres que se pueden emplear en las computadoras. Uno de los alfabetos más difundidos en la actualidad es el USA-ASCII, que consta de 256 caracteres. Los caracteres de que consta el alfabeto pueden ser letras, dígitos, signos de puntuación y símbolos especiales.

Alfanumérico, a *(Alphanumeric)* Relativo a cualquier carácter del alfabeto que use una computadora. (V. TEXTO ALFANUMÉRICO.)

ALGOL *(Algorithmic Language)* Lenguaje orientado al procedimiento y al cálculo. Es un lenguaje de alto nivel; fue creado en 1958 para ser usado por los matemáticos y modificado posteriormente para crear el ALGOL-60, así como el ALGOL-68.

Algoritmo *(Algorithm)* Descomposición en pasos u operaciones elementales de cualquier problema o suceso para su resolución o consecución óptima.

Almacenar *(Store)* Registrar, escribir, entrar datos en una computadora o en un soporte de almacenamiento externo de memoria.

Amplificador *(Amplifier)* Sistema o circuito que aumenta la potencia de una señal.

Analista *(Analyst)* Profesional de la computación que se dedica a planificar y analizar todos los procesos necesarios para solucionar un problema determinado o realizar una aplicación deseada mediante una computadora. Existen dos tipos de analistas: *analista de sistemas* y *analista de aplicaciones.*

AND Y. Operación lógica. Es la operación de multiplicar en el álgebra de Boole. Su comportamiento es el siguiente: A AND B será verdadero o 1 si A = verdadero o 1 y B = verdadero o 1, y falso o 0 cuando uno de los dos o los dos sean falsos o 0.

	0	1
0	0	0
1	0	1

ANSI *(American National Standards Institute)* Instituto Americano para Estándares Nacionales. Se trata de un instituto en el cual se dictan algunas normas que se convierten en estándares en el mundo de la computación. Por ejemplo, hay muchos lenguajes que cumplen las normas ANSI.

APL *(A programming language)* Lenguaje de programación de alto nivel inventado por K. Iverson en 1956. Se utiliza normalmente para cálculos matemáticos dado su carácter interactivo.

Aplicación *(Application)* Conjunto de programas creados para una computadora con el fin de agilizar diversos trabajos: contabilidad, gestión de almacenes, cálculos científicos, etcétera.

Archivo *(Archive)* Conjunto de datos almacenados según un rasgo común a todos ellos. (V. FICHERO.)

Área *(Area)* Cualquier zona de memoria que se usa para almacenar y transferir datos.

Argumento *(Argument)* Variable dependiente y asociada a una función. Su valor puede modificarse para obtener diferentes resultados de la función.

ASCII *(American Standard Code for Information Interchange)* Código normalizado americano para intercambios de información. Código internacional de 8 bits que permite a la computadora interpretar letras, dígitos, signos de puntuación u otros símbolos que se le introduzcan. Puede definir hasta 256 caracteres.

Asignación *(Assignement)* Acto de atribuir un valor a una zona de memoria que normalmente se denomina variable.

Asíncrono, a *(Asynchronous)* Ausencia de regulación temporal automática en la transmisión de datos o de infor-

mación entre varias computadoras o entre una de ellas y sus periféricos. La transmisión de datos puede ser regulada por el usuario mismo. Se opta por esta transmisión cuando se hace por vía telefónica normal o en transmisiones de baja velocidad.

Automatización *(Automation)* Proceso llevado a cabo por máquinas.

Autómata *(Automata)* Aparato con un mecanismo interno que le permite efectuar determinadas operaciones.

Autónomo, a *(Stand Alone)* Dícese de la máquina que funciona por sí sola, es decir, que no depende de otra máquina para su buen funcionamiento.

B

Banco de datos *(Data bank)* Conjunto de la información alrededor de un tema, disponible para su uso. Desde cualquier computadora se puede conectar con el banco de datos para requerir información. La información contenida en un banco de datos suele ser muy extensa.

Base de datos *(Data base)* Conjunto de ficheros de datos organizados según un método que facilita la actualización, acceso y recuperación de la información contenida en ellos. El uso de la base de datos se restringe a una o varias computadoras, mientras que el *banco de datos* es de uso general. Compárese, para entender la diferencia entre banco de datos y base de datos, el uso que hace un particular de una biblioteca pública (banco de datos) y la suya propia (base de datos).

BASIC *(Beginners All-purpose Symbolic Instructions Code)* Código de instruciones simbólicas de uso general para principiantes. Lenguaje de alto nivel, muy popular, sencillo en su manejo y fácil de enseñar y aprender. Es el lenguaje de programación más difundido en la actualidad.

Baudio *(Baud)* Unidad de medida de capacidad de transmisión de información equivalente a un bit por segundo. Una línea capaz de transmitir 1.200 baudios es capaz de transmitir 1.200 bits por segundo. El nombre procede de E. Baudot, su inventor, que utilizó un código de 5 bits para la transmisión de télex y teletipos.

Biblioteca *(Library)* Conjunto de programas o de rutinas archivadas en un dispositivo de almacenamiento externo de memoria, al cual se puede acceder desde un programa o desde un compilador.

Bidireccional *(Bidirectional)* Se aplica a todo dispositivo que permita la circulación de datos en cualquier dirección o sentido.

Bifurcación *(Jump)* Salto en la secuencia de órdenes del programa; se usa en programación. La bifurcación puede ser *condicional*, cuando depende del cumplimiento o no de una determinada condición, e *incondicional*, cuando no depende de nada.

Binario *(Binary)* Dícese de un elemento físico o lógico que puede presentar dos estados conceptualmente asimilables a presencia o ausencia. También se aplica al sistema de numeración que tiene por base el número 2. Las computadoras utilizan este sistema de numeración, porque no conocen más que los dos estados citados anteriormente, presencia o ausencia de corriente.

Bit *(Binary Digit)* Cifra binaria. Unidad mínima de información, que sólo puede tomar uno de los dos valores siguientes: 0 o 1. Si examinamos en detalle toda la información presente en un dispositivo de proceso de datos veremos que está constituida por bits. En el campo de la computación los bits se agrupan de 8 en 8 para formar *bytes*.

Blanco *(Blank)* Espacio. Carácter que representa el lugar que queda vacío entre dos palabras, por ejemplo, de un texto impreso.

Bloque *(Block)* Conjunto de bytes que pueden leerse o grabarse físicamente de una sola vez desde o en un dispositivo de almacenamiento externo.

BOF *(Beginning OF File)* Principio de fichero. Marca que indica el comienzo de un fichero, en un soporte de almacenamiento externo de memoria (discos, cintas, etcétera).

Borrar *(Clear)* Poner la pantalla en blanco. || Anular la información contenida en un fichero o en un registro. || Poner a cero una determinada zona de memoria.

BPI *(Bits Per Inch)* Bits por pulgada. Unidad de densidad de grabación sobre cinta o sobre disco.

Bucle *(Loop)* En programación, ejecución repetida de un determinado número de instrucciones de un programa hasta llegar al cumplimiento de una condición que anula esta reiteración y hace que se efectúe la siguiente instrucción que se encuentra después del bucle.

Buffer Área de memoria que almacena temporalmente la información de entrada o salida de cualquier dispositivo periférico (impresora, módem, diskette, hard disk).

Burótica Ver ofimática.

Byte Grupo de 8 bits con el que se representa un carácter. Las instrucciones en lenguaje máquina de una computadora ocupan 1, 2 o 3 bytes. Es la unidad de memoria más pequeña que puede almacenar la C.P.U. y su contenido puede ser cualquier número binario entre 00000000 y 11111111.

C

C Lenguaje de programación de alto nivel ideado por D. Ritchie. Fundamentalmente es un lenguaje dedicado al desarrollo de sistemas; aunque actualmente, como lenguaje fundamental del sistema operativo UNIX, se está empleando también para desarrollar aplicaciones para el usuario.

Cabezal *(Head)* Dispositivo electrónico que sirve para leer o grabar información de o en una unidad de almacenamiento externo (cinta, hard disk, diskette).

CAD *(Computer Aided Design)* Diseño asistido por computadora. Uso de las computadoras en el campo del diseño.

CAD/CAM *Computer Aided Design/Computer Aided Manufacturing)* Diseño asistido por computadora/Fabricación asistida por computadora. Aplicación de las computadoras en el campo del diseño y en el de la fabricación.

CAE *(Computer Aided Education)* Educación asistida por computadora. Aplicación de las computadoras en el campo de la educación como ayuda al maestro en su labor docente y al alumno en su aprendizaje.

CAM *(Contents Addressable Memory)* Memoria direccionable por contenido. (V. MEMORIA ASOCIATIVA.)

Canal *(Channel)* Vía de comunicación entre la unidad central de proceso (V. CPU) y los dispositivos de control de entrada/salida (unidades de disco, de diskette, impresoras).

Carácter *(Character)* Cualquier símbolo utilizado para representar letras, cifras, signos de puntuación, otros signos especiales, etc. Cada carácter ocupa un byte de memoria.

Cargador *(Loader)* Programa del sistema operativo que efectúa y controla la carga de programas en memoria.

Cargar *(Load)* Introducir cualquier tipo de información posible en la memoria de la computadora.

Cartucho *(Cartridge)* Soporte de un dispositivo de almacenamiento externo, normalmente una unidad de cinta.

Cassette Cartucho de cinta magnética de poca capacidad, que sirve para almacenar información. Es el soporte de un dispositivo de almacenamiento externo.

CBASIC *(BASIC Compiler)* Compilador BASIC. Compilador bastante difundido para microprocesadores 8080, Z80,

8085. La mayor parte de los lenguajes BASIC son interpretados y no compilados.

Cerrar *(Close)* Término genérico que se utiliza para indicar que se cierra un fichero. Tal acción se realiza siempre sobre un fichero abierto previamente, cuando ya no quiere utilizarse más.

Cibernética *(Cybernetics)* Término introducido en 1948 por Norbert Wiener y que titulaba su obra sobre problemas de comunicación y control entre los seres vivos y las máquinas. Su estudio abarca desde el comportamiento animal hasta el control de plantas industriales.

Cifra *(Digit)* Caracteres numéricos del 0 al 9. (V. DÍGITO.)

Cinta magnética *(Magnetic tape)* Tira de poliéster revestida de una emulsión magnética en la que se puede registrar información.

Clave *(Key)* V. LLAVE.

CLK *(Clock)* V. RELOJ INTERNO.

CMOS *(Complementary Metal Oxide Semiconductor)* MOS complementario. Familia de circuitos integrados en transistores de tecnología MOS, con muy poco consumo de energía y que combina la enorme densidad de integración de los transistores PMOS y la gran velocidad de los transistores NMOS.

COBOL *(Common Business Oriented Language)* Lenguaje común para actividades comerciales. Lenguaje de programación de alto nivel concebido para su aplicación en la programación comercial. Nació en 1959 y utiliza el inglés común.

Codificar *(Encode)* Traducir la información que se quiere introducir en la computadora a un lenguaje que ésta pueda interpretar.

Código *(Code)* Conjunto de símbolos convencionales que permiten representar los datos para su manejo en la computadora.

Código de barras *(Bar code)* Conjunto de rayas verticales u horizontales de distinto espesor que contiene información del producto sobre el que figuran. Se usa este código en productos de alimentación, revistas, etc., para informar a la caja registradora; también se usa como partitura para los sintetizadores.

Código fuente *(Source code)* Forma de redacción de cualquier tipo de lenguaje que, para su procesamiento, deberá sufrir el proceso de compilación. Las instrucciones en código fuente coinciden con las del lenguaje de programación con el que está hecho el programa.

Código Hollerith *(Hollerith code)* Código inventado por Hollerith para la representación de caracteres en tarjetas perforadas.

Código objeto *(Object code)* Resultado de la compilación o traducción de un lenguaje fuente.

Cola de espera *(Waiting queue)* Área de memoria donde se colocan los elementos de información que están a la espera de ser procesados. (V. FIFO y LIFO.)

Comando *(Command)* Galicismo usado con el sentido de orden. En el lenguaje informático los comandos se utilizan para ordenar al procesador que ejecute una función específica. Por ejemplo, en BASIC el comando LOAD significa la carga en memoria de un programa para su ejecución, que se consigue con el comando RUN.

Compatibilidad *(Compatibility)* Afinidad entre una computadora y sus periféricos, o entre diferentes computadoras que tienen el mismo código y pueden ejecutar el mismo software.

Compilado *(Compiled)* Relativo a los programas cuyas instrucciones no se encuentran en lenguaje máquina porque han sufrido un proceso de compilación.

Compilador *(Compiler)* Software o programa que realiza la traducción de un programa escrito en un lenguaje fuente de alto nivel (FORTRAN, COBOL, PASCAL, C, etc.) a lenguaje máquina directamente asimilable por el computador. La ejecución de los programas compilados es mucho más rápida que la de los programas interpretados. El único inconveniente que tienen los compiladores es que el código máquina que generan es mucho más voluminoso en memoria y más lento en ejecución que el código que habría producido un buen programa escrito en ensamblador.

Compilar *(Compile)* Acción que realiza el compilador.

Computación *(Computing)* Ciencia que estudia el manejo y tratamiento automático de la información mediante el uso de computadoras.

Computadora *(Computer)* Máquina básica para tratar de forma automática la información. Está formada por una unidad central de proceso, memorias y unidades de entrada/salida de datos. Las hay de varios tipos: de bolsillo, personales, etc. Se usa también para designar a estas máquinas el término en masculino **computador**, y también como sinónimo *ordenador*, del francés *ordinateur*.

Computadora doméstica *(Home computer)* Pequeña computadora, de bajo coste, útil para iniciarse en computación, y que, a pesar de su limitada capacidad de información, puede efectuar los pequeños trabajos de una oficina o de un laboratorio.

Computadora personal (P.C.) *(Personal computer)* Computadora intermedia entre la doméstica y la minicomputadora de coste medio, responsable en gran parte del auge actual de la computación. Su potencia ha ido en aumento y hoy puede gestionar unidades de almacenamiento externo de gran capacidad. Es un instrumento de trabajo útil para usuarios con profesiones liberales (médicos, abogados, etc.), para directivos de empresa, etcétera.

Configuración *(Configuration)* Forma de presentación de un equipo de proceso de datos (computadora y periféricos) referente a la presencia o no de discos, a las características de la unidad central, al tipo de impresora, a la clase y número de sus periféricos, etcétera.

Consola *(Console)* En las grandes computadoras, panel frontal de mandos e indicadores de una unidad, que sirve para que el operador dirija el sistema y controle su funcionamiento. En computadoras más pequeñas se considera una consola el grupo formado por pantalla y teclado.

Constante *(Constant)* Que no manifiesta variación a lo largo de la ejecución de un proceso.

Contador *(Counter)* Registro numérico o variable numérica que se utiliza en un programa para contar el número de veces que sucede una situación.

Control de paridad *(Parity check)* Control de la validez de una palabra o información recibida o leída; consiste en contar el número de bits con valor 1 leídos. Si el número es par el resultado será 0 o si es impar será 1. Este resultado debe coincidir con el bit de paridad de tal información; si no coincide, significa que la información no se ha recibido correctamente.

Conversacional *(Conversational)* Se dice del sistema (lenguaje o sistemas de otra índole) que permite un diálogo interactivo entre el usuario y la computadora a través de un terminal, generalmente la pantalla. V. LENGUAJE CONVERSACIONAL.

Convertidor *(Converter)* Dispositivo electrónico que permite la conexión de unidades diferentes. Así, un *convertidor de códigos* hace posible la conexión de unidades cuyo código es diferente; un *convertidor de señales* puede alterar el nivel de tensión de la entrada de las señales, etcétera.

CPI *(Characters Per Inch)* Caracteres por pulgada. Indica la densidad de impresión horizontal de una impresora. Es útil en tratamiento de textos.

CP/M *(Control Program for Microprocessors)* Programa de control para microprocesadores. Sistema operativo desarrollado en 1974 por el consejero de INTEL Gary Kildall, quien, en 1976, creó la Digital Research Corporation.

CP/M-86 Nueva versión del sistema operativo CP/M desarrollado por Digital Research Corporation y que funciona en computadoras que poseen como microprocesador el 8086 de INTEL de 16 bits. Es, junto con el S.O. MS-DOS, el sistema más difundido para computadoras personales. (V. CP/M.)

CPS *(Caracters Per Second)* Caracteres por segundo. Medida de velocidad de lectura/escritura de una memoria. Se le llama también B.P.S. (Bytes Per Second). (V. BAUDIO.)

CPU *(Central Processing Unit)* V. UNIDAD CENTRAL DE PROCESO.

CRT *(Cathode Ray Tube)* Tubo de rayos catódicos. Tubo electrónico del que están provistos muchos terminales y que les sirve para visualizar información. Es por lo tanto la pantalla de dichos terminales.

Cursor Raya o marca luminosa que indica el lugar que ocupará el carácter que se quiere visualizar en la pantalla de cualquier terminal de computadora. Tiene movilidad programada por toda la pantalla.

CH

Chip Pequeña cápsula de silicio que puede contener millones de circuitos integrados. Es capaz de memorizar datos o de gestionar información.

D

DAC *(Digital Analog Converter)* Convertidor digital analógico. Circuito electrónico que transforma representaciones digitales en señales analógicas.

Dato *(Data)* Unidad lógica de información que, junto con muchas otras, se suministra a un equipo para la resolución de una aplicación.

Datos analógicos *(Analog data)* Señales que representan las variaciones de las cantidades de las magnitudes físicas.

Datos digitales *(Digital data)* Información representada en forma digital (números, signos, símbolos, etcétera).

DBMS *(Data Base Management System)* Sistema de gestión de base de datos. Conjunto de programas que permiten almacenar y buscar datos en una base de datos de una manera sistemática.

Debugger Programa concebido para revisar y poner a punto los programas realizados para cualquier aplicación.

Decimal Sistema de numeración de base 10.

Decodificador *(Decoder)* Circuito electrónico que descifra los datos de entrada para que puedan ser utilizados por la computadora.

Decodificador de instrucciones *(Instruction decoder)* Dispositivo de la unidad central de memoria que descifra las instrucciones de un programa y las convierte en señales de mando para las partes de la CPU que intervienen en la ejecución de tales instrucciones: la unidad aritmética, los registros y la unidad control.

Decodificar *(Decode)* Acción inversa a la de codificar.

Decremento *(Decrement)* Valor constante que se resta a un contador, a un registro o a una variable, dentro de un programa.

Densidad *(Density)* Unidad de medida que sirve para medir la cantidad de información que puede grabarse en un soporte de almacenamiento externo de memoria por unidad de medida. Normalmente se expresa en bits por pulgada.

Digitalizador *(Digitizer)* Aparato que convierte señales analógicas en datos digitales. Se usa para efectuar gráficos o dibujos a partir de una información dada.

Digitalizar *(Digitize)* Conviene una señal en código digital; por ejemplo, transformar imágenes o sonidos en cifras.

Dígito *(Digit)* Cifra. Símbolo de una cantidad numérica. En el sistema de numeración binario hay dos dígitos: las cifras 0 y 1; en el decimal hay diez dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9.

Dirección *(Address)* Posición de una determinada información en una memoria o en un soporte, o sea, el valor numérico o alfanumérico con el que se puede identificar un dato contenido en la memoria o en un soporte magnético. Constituye un concepto fundamental en computación.

Directiva *(Directive)* V. PSEUDOINSTRUCCIÓN.

Directorio *(Directory)* Catálogo o índice que está registrado en un dispositivo de almacenamiento externo y que incluye los programas o ficheros de tal almacenamiento para facilitar su acceso al usuario.

Disco *(Disk)* Dispositivo magnético rotatorio en forma de disco (de ahí su nombre) que almacena información en sus surcos o pistas concéntricas. Hay dos clases distintas de discos: el *disco duro* (hard disk) y el disco *flexible* (floppy disk o diskette). Los hay *fijos*, utilizados como extensión de la memoria central, e *intercambiables*, cuya presentación en cartuchos permite sacarlos o introducirlos tantas veces como sea necesario de la unidad que soporta.

Disco duro *(Hard disk)* Disco magnético de un material rígido (aluminio, por ejemplo), utilizado para almacenar datos. Posee mayor capacidad que un diskette o disco flexible.

Diskette Disco magnético flexible de pequeñas dimensiones usado para almacenar programas u otra información. Se le llama también *floppy disk*. (V. DISCO.)

Display Periférico, generalmente una pantalla, donde aparece de forma visual la información requerida.

DMA *(Direct Memory Access)* Acceso directo a memoria. Método mediante el cual es posible grabar u obtener datos directamente de la memoria, es decir, sin pasar por la unidad central de proceso (CPU). Se utiliza para realizar transferencias entre memoria y periféricos y al revés. De este modo los tiempos de transferencia son menores.

E

EBCDIC *(Extended Binary Coded Decimal)* Código decimal binario de 8 bits, usado para la representación de caracteres. Su extensión es de 256 caracteres, al igual que el código ASCII, del que sólo se diferencia por el orden de estos caracteres.

Editar *(Edit)* Tratar un texto (componerlo, modificarlo, etc.) a través de la pantalla de la computadora utilizando un programa que normalmente se denomina editor.

Editor *(Editor)* Programa que permite editar textos o programas a través de la pantalla de vídeo.

Ejecutar *(Execute)* Referido a programas o aplicaciones, segnifica el proceso normal de este programa o aplicación por parte de la computadora.

Emulador *(Emulator)* Programa o dispositivo especial que es capaz de traducir instrucciones del lenguaje de otra computadora al lenguaje de aquella en que está instalado el equipo emulador. De esta manera ciertas computadoras pueden utilizar programas destinados a otras, e incluso simular el comportamiento de otra.

En línea *(On line)* Se dice que un terminal inteligente está en línea cuando está funcionando conectado directamente con la unidad central de proceso.

Encadenamiento *(Chaining)* Encadenamiento de programas: Posibilidad de ordenar la ejecución de un programa como instrucción del que se está ejecutando.

ENIAC *(Electronics Numerical Integrator and Calculator)* Una de las primeras computadoras, construida en 1964; funcionó en la universidad de Pennsylvania desde 1946 a 1955.

Enmascarar *(Mask)* Acción que se consigue a través de una operación lógica y que consiste en extraer de un conjunto de caracteres un grupo seleccionado de ellos.

Ensamblador *(Assembler)* Software o programa que traduce programas escritos en lenguaje simbólico de bajo nivel a lenguaje máquina, que es el único asimilable por la computadora. (V. LENGUAJE ENSAMBLADOR.)

Entrada *(Input)* Introducción de datos en una computadora, ya sea mediante el teclado o mediante cualquier periférico con el que se pueda realizar. El término se aplica también a los dispositivos que se usan para tal fin.

EQ *(Equal)* Igual a. Operador lógico que se utiliza en la mayoría de lenguajes.

Ergonómico Dícese del dispositivo que ha sido especialmente diseñado para hacer agradable su manejo por parte del usuario; así, *pantalla ergonómica* es aquella que no perjudica, por ejemplo, la vista.

Etiqueta *(Label)* Carácter o grupo de caracteres que identifican una sentencia de un programa a la cual se accederá a través de otra sentencia. || Cada uno de los nombres de todos los ficheros almacenados en una unidad de almacenamiento externo.

F

FD *(Floppy disk)* V. DISKETTE.

Fibra óptica *(Optical fiber)* Hilos muy finos de sílice o de materia plástica transparente con un índice de reflexión muy elevado y que se encuentran dentro de un envoltorio del mismo material, pero con un índice de refracción lo suficientemente débil como para impedir la salida de luz. Por ellos circulan señales luminosas, generadas a partir de señales eléctricas en un extremo; en el otro, tales señales se convierten otra vez en eléctricas.

Ficha *(Card)* V. TARJETA PERFORADA.

Fichero *(File)* Conjunto de datos o instrucciones almacenados en un soporte magnético. Un programa es también un fichero. La utilización aislada de los ficheros y su acceso requieren un buen sistema de tratamiento de ficheros.

FIFO *(First Input First Output)* Técnica empleada en valoración de almacenes o en una cola de espera, según la cual el primer artículo o información almacenados es el primero en salir o la primera en procesarse, respectivamente.

Floppy disk *(Diskette)* Disco flexible, recubierto de óxido magnético, que gira en el interior de una envoltura que limpia su superficie. Existen tres tamaños de discos flexibles: de 8 pulgadas (20 cm de diámetro), de 5,25 pulgadas (13 cm) y de 3,5 pulgadas.

Formatador *(Formater)* Programa que permite formatar los soportes de las unidades de almacenamiento externo.

Formato *(Format)* Estructura que se aplica a los soportes de las unidades de almacenamiento externo según el criterio del usuario y las posibilidades de la máquina para así poder trabajar con ellos.

FORTH Lenguje de programación creado por Charles H. Moore. Es un lenguaje parecido al BASIC, menos fácil de aprender, aunque más rápido en la ejecución de programas.

FORTRAN *(Formula Translation)* Traductor de fórmula. Lenguaje de programación orientado a problemas planteados en forma matemática. Se trata de un lenguaje avanzado, pero actualmente ya anticuado, que fue concebido para cálculos numéricos. Es el «padre» del BASIC.

Fuente de alimentación *(Power supply)* Dispositivo que transforma la energía eléctrica procedente del exterior al voltaje necesario para los circuitos del sistema.

Fuera de línea *(Off line)* Modo de operar con autonomía, por ejemplo un terminal en funcionamiento cuando está desconectado de la CPU. (En jerga informática se usa comúnmente el termino inglés.)

Full duplex Medio de comunicación que permite la transmisión simultánea e independiente de datos en las dos direcciones de una conexión entre dos puntos.

G

Generador *(Generating)* Que crea o produce automáticamente.

Generador de listados *(Report generating)* Programa que permite obtener listados de diversos tipos tomando siempre los datos de un archivo.

Generador de programas *(Program generating)* Programa que produce a partir de una información determinada el programa correspondiente.

Generar *(Generate)* Crear o producir algo mediante una máquina.

Gigabyte Gigaocteto. Equivale a 10^9 bytes o, lo que es lo mismo, mil millones de octetos.

Grabar *(Write)* Escribir información en una unidad de almacenamiento externo.

GT *(Greater Than)* Mayor que. Operador lógico que se utiliza en la mayoría de lenguajes.

H

Hard copy Copia permanente. Se donomina hard copy a la copia exacta sobre papel de la información que aparece en la pantalla.

Hardware Conjunto de componentes físicos (cables, tornillos, placas, etc.) que constituyen una computadora.

Hexadecimal Sistema de numeración de base 16. Se utilizan las cifras del 0 al 9 seguidas de las 6 primeras letras del alfabeto, para representar los números de 0 al 15. Normalmente las cifras hexadecimales se escriben con una H al final, para no confundirlas con una cifra decimal.

HP Hewlett-Packard. Una de las compañías punteras en la fabricación de computadoras, con sede en Fort Collins, Colorado (USA). Debe su nombre a sus dos fundadores.

Hz Hertz. Unidad de medida de frecuencia.

I

IBM *(International Business Machines)* Corporación internacional de máquinas de gestión. Compañía americana constructora de computadoras que ha aportado importantes innovaciones en el campo de la computación.

Impresora *(Printer)* Uno de los periféricos exclusivamente de salida más importantes. Existen varios tipos de impresoras según la calidad y la velocidad de impresión, la naturaleza de los caracteres, el soporte del papel, etc. Algunas solamente imprimen cifras y letras, y otras son mixtas, o sea, alfanuméricas y gráficas. A las impresoras exclusivamente gráficas se las llama *plotter*. Según la cantidad de información que pueden imprimir de una sola vez, se dividen en: *impresoras de carácter* (impresión carácter a carácter), *impresoras de línea* (impresión línea a línea de una sola vez) e *impresoras de página* (composición de una página entera en cada impresión).

Impresora bidireccional *(Bidirectional printer)* Máquina impresora que puede imprimir datos tanto de derecha a izquierda como de izquierda a derecha.

Impresora de barras *(Bar printer)* Máquina que imprime caracteres alfanuméricos y otros especiales dispuestos en una barra que se va desplazando según el sentido de la impresión. Su velocidad de impresión puede ser de 600 líneas por minuto.

Impresora de cadena *(Chain printer)* Máquina en la cual los caracteres que hay que imprimir van colocados en una cadena que se desplaza horizontalmente a una velocidad de 1.600 líneas por minuto.

Impresora de chorro de tinta *(Ink jet printer)* Modelo de impresora de línea en la que los caracteres se van formando mediante pequeños chorros de tinta sobre el papel. Los caracteres se forman a base de puntos de tinta.

Impresora de láser *(Laser printer)* Impresora que emplea para la impresión de datos la técnica del láser.

Impresora de margarita *(Daisy wheel printer)* Máquina en la que los caracteres están sobre un soporte circular en forma de tulipán o margarita.

Impresora de matriz de agujas *(Matrix printer)* Máquina que imprime caracteres en forma de puntos por medio de unas agujas impresoras. Su velocidad es de 1.200 líneas por minuto. Puede ser bidireccional o unidireccional.

Impresora térmica *(Thermal printer)* Impresora en la que la impresión no se realiza por presión y entintado del papel, sino por quemado de un papel tratado especialmente. Se trata de máquinas muy silenciosas.

Incremento *(Increment)* Valor constante que se suma a un contador, a un registro o a una variable, dentro de un programa.

Indexación *(Indexing)* Supongamos que tenemos un fichero A cuyos registros no están clasificados por ninguna clave. Si quisiéramos obtener los registros de este fichero A clasificados por alguna clave o por alguno de sus campos, deberíamos construir otro fichero B en el cual grabaríamos todas las claves clasificadas y, además, por cada clave grabaríamos la dirección del registro que la contiene en el otro fichero. A la construcción de este segundo fichero se la denomina indexación.

Indexado *(Indexed)* Dícese del fichero cuyos registros no se encuentran clasificados y que se desea obtener su categorización mediante el proceso de indexación.

Indicador *(Indicator)* Dispositivo utilizado para atraer la atención. En programación se utiliza para señalar que se ha efectuado una determinada operación para más tarde comprobarlo.

Índice *(Index)* Se llama así al fichero que se construye mediante el proceso de indexación.

Indirecto *(Indirect)* Modo de direccionamiento utilizado en programación cuando la posición de memoria que hay que leer o escribir no se encuentra en la propia instrucción, sino en otra zona de memoria o registro cuya dirección sí está indicada en la instrucción.

Informática *(Computing)* V. COMPUTACIÓN.

Inicialización *(Initialization)* Adecuación de todos los elementos de un sistema para su puesta en marcha o para poder trabajar con él.

Insertar *(Insert)* Incluir un nuevo elemento dentro de un conjunto de ellos y en un lugar determinado. Por ejemplo, se puede insertar una palabra dentro de un texto o una letra dentro de una palabra, durante la composición de textos con una computadora.

Instrucción *(Instruction)* En lenguajes de bajo nivel, operación elemental reconocible y ejecutable por la unidad central de proceso de una computadora. En lenguaje de alto nivel es una frase o sentencia que puede desglosarse en varias operaciones elementales también reconocibles y ejecutables por la unidad central de proceso. En computación es sinónimo de orden.

Inteligencia artificial *(Artificial intelligence)* Concepto bajo el cual se engloban todas las tecnologías que estudian la creación de máquinas (robots, autómatas, etc.), y también todos los programas que se ejecutan siguiendo un método parecido a la inteligencia humana; por ejemplo, traducciones de una lengua a otra, juegos de ajedrez, etcétera.

Interactivo *(Interactive)* V. CONVERSACIONAL.

Interface Es el medio físico y lógico común y necesario de dos sistemas para intercambiar comunicación. Así, una computadora envía datos a una impresora a través de una interface.

Intérprete *(Interpreter)* Programa que traduce y ejecuta a lenguaje máquina cada una de las instrucciones de un programa escrito en un lenguaje de alto nivel.

I/O *(Input/Output)* Entrada/Salida. Elementos físicos que componen una vía por donde circula la información que la computadora suministra a sus periféricos o viceversa.

IPS *(Inches Per Second)* Pulgadas por segundo.

IRS *(Interrupt Request Signal)* Señal de petición de interrupción. Señal que se envía a la CPU para que ésta efectúe o no la interrupción ordenada por un programa o un proceso.

ISAM *(Indexed Sequential Access Method)* Método de acceso secuencial indexado. Método utilizado en muchos paquetes que sirven para construir un fichero índice con el que se puede tener clasificado constantemente un fichero de datos. El acceso al fichero de datos se realiza siempre pasando por el fichero índice.

Iterar *(Repeat)* Repetir una instrucción u operación un determinado número de veces hasta que no se cumpla una determinada condición.

J

JCL *(Job Control Language)* Lenguaje de control de trabajos o lenguaje de órdenes. Término que se aplica a todos los lenguajes que instruyen sobre las tareas que el sistema operativo de una computadora debe realizar.

Juego de caracteres *(Character set)* Gama o conjunto de caracteres que maneja una computadora. Conjunto de tipos de una impresora.

Juego de instrucciones *(Instruction set)* Conjunto de instrucciones ejecutables por una computadora, Normalmente, dicho juego está formado por instrucciones aritméticas, lógicas, de test y de transferencia de información.

Jump Salto. (V. BIFURCACIÓN.)

Justificación *(Justification)* Encuadre de un texto. Este encuadre se puede hacer a la derecha y a la izquierda.

K

K Múltiplo del byte, equivalente a 2^{10} bytes, o sea, 1.024 bytes. También se le llama Kbyte o Kilobyte.

Kilobaudio *(Kilobaud)* Mil baudios o mil bits por segundo. Es la unidad que mide la frecuencia de transmisión de datos.

Kilobit Múltiplo del bit, equivalente a 1.024 bits.

Kilobyte Kiloocteto. Unidad de medida de memoria central y de dispositivos de almacenamiento externo. Equivale a 1.024 bytes.

Kit Sistema presentado por piezas montables por el propio usuario. Con esta presentación se abarata el producto.

L

Lápiz óptico *(Light pen)* Dispositivo que puede actuar de periférico de entrada de datos a la computadora. Puede sustituir al teclado si el usuario lo prefiere. Va provisto de una célula fotoeléctrica que hace que la computadora reconozca los caracteres cuando el lápiz los enfoca.

Lector *(Reader)* Dispositivo periférico que recoge información de un soporte (papel, cinta magnética) y la introduce en la computadora para su tratamiento.

Lector de caracteres magnéticos *(Magnetic reader)* Dispositivo que permite leer caracteres magnéticos contenidos en áreas magnetizadas de fichas o documentos de papel.

Lector de cinta de papel *(Paper tape reader)* Lector que reconoce la información registrada mediante perforaciones en una cinta de papel. Según la forma de detectar la información, pueden ser eléctricos o fotoeléctricos.

Lector de cintas magnéticas *(Magnetic tape reader)* Lector que puede leer la información contenida en cintas magnéticas, cassettes, etcétera.

Lector de código de barras *(Bar code reader)* Lector que reconoce los datos contenidos en una combinación de rayas horizontales o verticales de diferente grosor impresas sobre un soporte de papel.

Lector de tarjetas *(Card reader)* Periférico que identifica la información contenida en un soporte (la tarjeta) por las perforaciones que hay en él. (También llamado *lector de fichas.)*

Lector óptico de caracteres *(Optical mark reader)* Lector capaz de identificar marcas impresas, generalmente caracteres alfanuméricos, sobre papel.

Lectura *(Read)* Acción por la que un dispositivo obtiene información de una memoria interna o externa.

Lectura de instrucción *(Fetch)* Primera fase de toda ejecución de una instrucción. La ejecución de una instrucción tiene dos fases: una de lectura de la instrucción y otra de realización de lo que en esta lectura se ordena.

Lectura de memoria *(Memory read)* Acto por el que se extrae información contenida en la memoria.

Lectura de pantalla *(Screen read)* Acto por el cual se tansmite a un periférico la información que se visualiza en la pantalla de la computadora.

Lenguaje binario *(Binary language)* V. LENGUAJE MÁQUINA.

Lenguaje compilador *(Compiler language)* Cualquier tipo de lenguaje simbólico concebido para facilitar al usuario la tarea de programación, y que no es ejecutable directamente por la computadora, sino después de un proceso de traducción.

Lenguaje conversacional *(Conversational language)* Lenguaje interactivo, es decir, que permite el diálogo entre el usuario y la computadora.

Lenguaje de alto nivel *(High-level language)* Lenguaje evolucionado. Lenguaje de programación simbólico, parecido a las lenguas naturales, independiente por tanto del lenguaje máquina, lo que facilita su manejo y su aprendizaje.

Lenguaje de bajo nivel *(Low-level language)* Lenguaje de programación poco evolucionado, próximo al lenguaje máquina, en el que a cada instrucción le corresponde otra en lenguaje máquina.

Lenguaje de programación *(Programming language)* Sistema de signos y símbolos que, mediante un conjunto de reglas, permite la construcción de programas con los que la computadora puede operar. Los hay de alto nivel APL, BASIC, FORTRAN y de bajo nivel ASSEMBLER (ensamblador).

Lenguaje de programación concurrente *(Concurrent programming language)* Lenguaje de programación para programas concurrentes; es decir, que se pueden ejecutar con simultaneidad. (V. MULTIPROGRAMACIÓN.)

Lenguaje ensamblador *(Assembler language* o *Assembly language)* Lenguaje de programación de bajo nivel en el que a cada instrucción simbólica le corresponde una instrucción máquina. Se diferencia del lenguaje máquina porque su código está representado por signos mnemotécnicos.

Lenguaje estructurado *(Structured language)* Tipo de lenguaje creado para facilitar la programación estructurada.

Lenguaje fuente *(Source language)* V. CÓDIGO FUENTE.

Lenguaje máquina *(Machine language)* Lenguaje de más bajo nivel. el único que la computadora comprende. Usa el código binario, de ahí que se le llame también *lenguaje binario*. Es el único que se corresponde con el funcionamiento de la computadora, por lo que permite explotar al máximo la capacidad del equipo; sin embargo, es muy lento y complejo programar en este lenguaje.

Lenguaje objeto *(Object language)* V. CÓDIGO OBJETO.

Lenguaje orientado al problema *(Problem oriented language)* Cualquier lenguaje creado para la resolución de unos problemas determinados; por ejemplo, el lenguaje BASIC.

Lenguaje orientado al procedimiento *(Procedure oriented language)* Cualquier lenguaje diseñado para resolver unos determinados tipos de aplicaciones; por ejemplo, el lenguaje FORTRAN o el COBOL: el primero para aplicaciones científicas y el segundo para aplicaciones de gestión comercial.

Lenguaje simbólico *(Symbolic language)* Cualquier lenguaje más evolucionado que el lenguaje máquina. Las instrucciones de un lenguaje simbólico, así como las direcciones, se expresan nominalmente en lugar de usar un código númerico, como es el caso del lenguaje máquina.

LIFO *(Last-In-First-Out)* Última entrada-primera salida. Técnica que se aplica muy a menudo en valoración de almacenes y que sigue la norma de que los lotes que van saliendo de almacén se valoran al precio de los más recientemente entrados.

Limpiar *(Clear)* V. BORRAR.

Línea *(Line)* Unidad con la que normalmente se mide un programa. Así, se dice que un programa consta de 100 líneas, 1.000 líneas, etc. Una línea de programa puede contener más de una instrucción.

LISP Lenguaje de programación de alto nivel, creado por J. MacCarthy, especializado en el tratamiento de listas. Es un lenguaje interactivo, bastante complejo, que se emplea sobre todo en inteligencia artificial (AI). No está muy difundido a causa de su complicada sintaxis.

Listado *(Listing)* Cualquier representación impresa producida por una computadora.

Listado ensamblado *(Assembly listing)* Listado de un programa hecho en *Assembler* y generado por el ensamblador. La característica principal de este listado consiste en que las instrucciones no están en lenguaje fuente, sino en lenguaje máquina.

Literal Todo símbolo o conjunto de símbolos que representan una constante. Esta constante puede encontrarse representando un dato en el interior de una instrucción.

Lógica booleana *(Boolean logic)* Parte de la lógica, adjetivada con el nombre de su creador, George Boole, en la que se tratan las operaciones lógicas binarias fundamentales AND, OR y NOT; a partir de ellas se definen otras más complicadas.

Logical *(Software)* V. SOFTWARE.

LOGO Lenguaje de programación para la enseñanza, que recurre esencialmente a los gráficos. Se considera un lenguaje de alto nivel, pero al mismo tiempo es lo suficientemente sencillo para que puedan aprenderlo hasta los niños

de corta edad. Además de enseñar a los niños a programar, amplía enormemente los conceptos geométricos.

Longitud de registro *(Record lenght)* Número de caracteres o bytes que forman un registro lógico o físico.

Longitud de una instrucción *(Instruction length)* Número de bytes que forman una instrucción.

LPS *(Lines Per Second)* Líneas por segundo. Es la expresión de la velocidad de impresión de una impresora de alta velocidad.

LSI *(Large Scale Integration)* Integración a gran escala. Sistema de integración de circuitos mediante el cual se pueden conseguir de 1.000 a 10.000 componentes.

LT *(Less Than)* Menor que. Siglas que en algunos lenguajes se utilizan como operadores aritméticos de comparación.

LL

Llamada *(Call)* (Normalmente **llamada** a un subprograma desde otro programa). Instrucción de un programa de modo que, al ejecutarse, hace que la secuencia de ejecución del programa varíe y se transfiera el control a otra zona de memoria donde se encuentra el subprograma. Este subprograma será ejecutado y, una vez finalizada su ejecución, el control volverá a la siguiente instrucción, a la *call,* del programa principal.

Llamada condicional *(Conditional call)* Llamada normal *(call)* en la cual la transferencia del control al subprograma solamente se efectúa si se cumple una determinada condición.

Llave *(Key)* Carácter o grupo de caracteres que se utilizan para poder identificar cada uno de los registros lógicos de un fichero.

M

Macroensamblador *(Macroassembler)* Ensamblador que permite el uso de macroinstrucciones.

Macroinstrucción *(Macroinstruction)* Instrucción en lenguaje simbólico, que, después de pasar por el proceso de compilación, se transforma en una secuencia de instrucciones máquina.

Macrosentencia *(Macrostatement)* Sentencia que dentro de un programa engloba otras sentencias del mismo lenguaje en el que está escrito tal programa.

Mailing *(Mailing)* Impresión automática de cartas con el membrete personalizado a partir de un fichero de nombres y direcciones. Existen programas que realizan esta función y que normalmente forman parte de un paquete de proceso o tratamiento de textos.

Manipulador *(Handler)* Programa que se utiliza para controlar un periférico o para comunicarse con él. Existe un manipulador para cada periférico. (También se le llama *manejador*.)

Mantenimiento *(Maintenance)* Conjunto de operaciones que se efectúan sobre una computadora para conservarla en perfecto estado. Estas operaciones afectan tanto al hardware como al software.

Mantisa *(Mantissa)* Parte decimal de una cantidad expresada en coma flotante.

Mapa de memoria *(Memory map)* Relación de las direcciones asignadas a las instrucciones y operandos de programas que se encuentran en una memoria. No es nada más que la representación en hexadecimal de todas los datos que hay en memoria.

Máquina *(Machine)* Procesador, computadora.

Marca *(Mark)* Símbolo o carácter utilizado para representar el fin de una zona de datos.

Mark 1 Máquina de calcular, antecesora de todas las computadoras. Pesaba dos toneladas, empleaba 850 kilómetros de cables eléctricos y efectuaba 200 operaciones por minuto.

Máscara *(Mask)* En lógica de computación, conjunto de bits que se utiliza para transformar otro conjunto de bits (para ponerlos a 0 o a 1). En programación, conjunto de caracteres que, siguiendo unas normas específicas, hace que una cantidad se imprima con un formato determinado.

Matriz *(Array)* Cuadrado de dos dimensiones en el que las entradas horizontales son las líneas y las entradas verticales son las columnas.

Mega Abreviatura de Megabyte.

Megabyte Medida de almacenamiento expresada en bytes. Un Megabyte es igual a 1.000.000 de bytes.

Megaciclo *(Megacycle)* Medida para mesurar los ciclos. Se corresponde a 1.000.000 de ciclos.

Megahercio *(Megahertz)* Es la unidad de medida del ciclo básico de máquina.

Megaocteto *(Megabyte)* V. MEGABYTE.

Memoria *(Memory)* Dispositivo o parte de un equipo, destinado a almacenar de forma temporal o permanente informaciones codificadas y a devolverlas cuando se soliciten. En una computadora, la memoria se divide en dos partes: una memoria electrónica rápida, integrada al ordenador, y una memoria externa más lenta, compuesta por unidades de disco, cinta, etc. La capacidad de memoria se expresa en bytes.

Memoria alterable *(Alterable memory)* Memoria que permite cambiar los datos que contiene mediante operaciones de escritura.

Memoria anexa *(Bump)* Memoria a la que el programador no puede acceder directamente.

Memoria asociativa *(Contents addressable memory)* Memoria en la que se llega a una posición determinada no a través de su dirección, sino de su contenido.

Memoria auxiliar *(Auxiliary memory)* Almacenamiento que no forma parte de la unidad central, pero que es utilizado como soporte de esa unidad y que está permanentemente en línea. Unidades de memoria auxiliar pueden ser diskettes, discos, cassettes, etcétera.

Memoria central *(Central memory)* Memoria principal de una computadora.

Memoria de acceso directo *(Random access memory)* Tipo de memoria de lectura/escritura en la que el acceso a una posición no depende de la información anterior o posterior a la requerida. La memoria interna de una computadora es de acceso directo.

Memoria de acceso secuencial *(Sequential access memory)* Tipo de memoria de lectura/escritura en la que, para encontrar un dato, hay que pasar por toda la información registrada que le precede. Unidades de memoria de acceso secuencial son las cintas magnéticas, los cassettes, etcétera.

Memoria de burbujas *(Bubble memory)* Memoria de gran capacidad de almacenamiento por su técnica. Cada bit se almacena en una burbuja o pequeñísima zona magnetizada sobre una base de aluminio.

Memoria de disco *(Disk memory)* Almacenamiento de la información en discos magnéticos o diskettes.

Memoria de masa *(Mass memory)* V. MEMORIA AUXILIAR.

Memoria de sólo lectura *(Read only memory)* Memoria separada de la memoria central. La información que contiene es permanente, no se puede modificar, sólo leer. Su confi-

guración técnica permite una gran rapidez de acceso, por lo que se usa para contener programas o rutinas estándares, unidades de compilación, etcétera.

Memoria de trabajo *(Work memory)* Zona de la memoria central de una computadora destinada, mediante un programa, a contener datos que servirán para obtener los resultados requeridos. El establecimiento de esta memoria no es físico, sino lógico.

Memoria de usuario *(User memory)* Zona de la memoria central que, una vez cargado el sistema operativo, queda libre para albergar programas o datos con los que pueda trabajar el usuario.

Memoria del sistema *(System memory)* Zona de la memoria central reservada para albergar el sistema operativo y sus áreas de trabajo. Esta zona no es de un tamaño determinado, sino que varía según el tamaño del sistema operativo que se cargará en tal memoria.

Memoria dinámica *(Dynamic memory)* Memoria en la que la información se mueve físicamente y, como consecuencia, a veces no se encuentra disponible de inmediato.

Memoria estática *(Static memory)* Memoria en la que la información permanece inmóvil, desde un punto de vista mecánico, por lo que se accede a ella en un tiempo muy corto.

Memoria fugaz *(Volatile memory)* V. MEMORIA VOLÁTIL.

Memoria permanente *(Non volatile memory)* Memoria cuya información permanece, a pesar de desconexiones o fallos, en la alimentación de la computadora.

Memoria programable *(Programmable memory)* V. MEMORIA ALTERABLE.

Memoria programable de sólo lectura *(Programmable read only memory)* Memoria ROM que, mediante un aparato programador, puede ser programada según las propias rutinas del usuario, pero una sola vez; es decir, una vez programada no admite una nueva programación.

Memoria virtual *(Virtual memory)* Conjunto de software y hardware que consigue efectos de expansión de memoria por medio de un traductor dinámico de direcciones y una tabla de almacenamiento de direcciones reales y virtuales. La memoria virtual actúa como si ampliase la capacidad de la memoria central. La memoria se encuentra dividida en páginas, y solamente se encuentran en memoria central las páginas con las que realmente se está trabajando.

Memoria volátil *(Volatile memory)* Tipo de memoria que, cuando se desconecta la alimentación de la computadora, pierde toda la información que contenía. La mayoría de computadoras poseen memoria de este tipo.

Mensaje de error *(Error message)* Mensaje que dirige la computadora al programador o usuario, en el cual le describe una anomalía producida en el proceso que se estaba ejecutando.

Menú *(Menu)* Lista de opciones que un programa, al ser ejecutado, ofrece al usuario para que éste elija una de ellas. Una aplicación, al estar compuesta por muchos programas, ofrece varios menús; esto hace que la computadora y el usuario entren en un ambiente conversacional. El usuario elige el proceso que quiere realizar, la computadora lo ejecuta, y, una vez finalizado, ésta vuelve a mostrarle el menú de opciones para que el usuario vuelva a escoger. Una de las opciones siempre será el fin del trabajo.

Método de acceso *(Access method)* Modo a través del cual se accede a los datos o a la información contenidos en los dispositivos de almacenamiento externo y, en general, a cualquier memoria.

Mezcla *(Merge)* Fusión de varios grupos de datos para formar un solo grupo. Estos grupos pueden ser ficheros, matrices y programas.

Microcomputadora *(Microcomputer)* Computadora cuya unidad central de proceso es un microprocesador. Tiene capacidad limitada y puede tener conectados muy pocos periféricos.

Microelectrónica *(Microelectronics)* Rama o parte de la electrónica que estudia todo lo relativo a circuitos integrados.

Microprocesador *(Microprocessor)* Circuito integrado, comúnmente llamado chip, con integración a gran escala (LSI o VLSI); es la unidad central de proceso en una microcomputadora. Los circuitos integrados más reconocidos son los que fabrican las compañías INTEL, ZILOG y MOTOROLA.

Microprograma *(Microprogram)* Programa cuyas instrucciones son operaciones internas elementales ejecutadas por la unidad de control de una computadora al efectuar cada instrucción del programa de un usuario. Así si la instrucción de un programa de usuario es la multiplicación de dos números, la unidad de control para ejecutar esta multiplicación tendrá que seguir una serie de pasos elementales que conseguirá ejecutar a través del microprograma correspondiente a la instrucción de multiplicar. De ahí que podamos decir que existe un microprograma para cada instrucción elemental.

Microprogramación *(Microprogramming)* Técnica que se utiliza para la realización de microprogramas que serán ejecutados por la unidad de control de una computadora.

Microsegundo *(Microsecond)* Unidad de medida de tiempo. Su valor es 10^{-6} segundos.

Microsoft Compañía americana, puntera en el campo del desarrollo del software, sobre todo en el desarrollo del software de sistemas. Es famosa principalmente por su «intérprete BASIC», por su sistema operativo «MS-DOS» y por su programa aplicativo «Multiplan».

Milisegundo *(Milisecond)* Unidad de medida de tiempo que equivale a 10 segundos. En computación se usa para medir el tiempo que tarda una unidad de almacenamiento externo en leer una determinada información, en desplazar un cabezal, etcétera.

Minicomputadora *(Minicomputer)* Computadora de capacidad media, más pequeña que las grandes computadoras y más grande que las microcomputadoras. Su funcionamiento se asemeja más a una de las grandes que a una microcomputadora.

Minidisco *(Minidisk)* Disco magnético de pequeñas dimensiones, usado para almacenar datos u otra información.

Minidiskette *(Mini floppy disk)* V. DISKETTE.

Módem *(Modem)* Palabra formada por la primera sílaba del término modulador y las tres primeras letras de la palabra acumulador. Dispositivo de entrada/salida que, en comunicaciones, sirve para demodular la señal que llega de la línea o modular la señal que aparece en la computadora. Convierte las señales digitales que se producen en la computadora en señales analógicas aptas para viajar por la línea telefónica; también efectúa la función inversa, es decir, convierte las señales analógicas que viajan por la línea telefónica en señales digitales que la computadora puede interpretar.

MODULA Lenguaje de programación de alto nivel que permite la multiprogramación.

Modulador/demodulador *(Modem)* V. MÓDEM.

Módulo *(Module)* Cada uno de los elementos de un equipo, programa o proceso que son identificados de manera individual. Así, si un programa está dividido en módulos, se pueden ir cargando estos módulos en memoria para ser ejecutados uno tras otro. También se llama módulo a cada uno de los componentes de una computadora. Por ejemplo, la pantalla, la unidad central, el teclado, etc., serían módulos de la computadora.

Monitor *(Monitor)* Programa que forma parte del sistema operativo de una computadora. También puede denominarse «supervisor». Su función es la de controlar la continuidad de los trabajos, el manejo de los dispositivos de entrada/salida, la asignación de prioridades a los diferentes trabajos y, en resumen, la supervisión de todos los trabajos del usuario en estrecha relación con los sistemas internos (control de interrupciones, memoria, etcétera). || Pantalla de la computadora donde se visualiza la información de entrada/salida.

MOS *(Metal Oxyde Semiconductor)* Semiconductor de óxido metálico. Tecnología usada para la fabricación de circuitos integrados de integración a gran escala. Su nombre procede de las tres iniciales de un metal (aluminio), un óxido de sílice y un semiconductor de silicio, que constituyen las tres capas de las que se compone un transistor o un chip construido con esta tecnología.

MP/M *(Multiprogramming Control Program for Microprocessors)* Programa de control de multiprogramación para microprocesadores. Sistema operativo que soporta multiprogramación y que es una versión del sistema operativo CP/M de Digital Research Corporation.

Muestreo *(Sampling)* Examen de algunos elementos de un conjunto para poder extraer datos que se supongan indicativos de algunas características del conjunto.

Multiplexado *(Multiplexing)* Método de transmisión por línea de comunicaciones que consiste en la transmisión de varias señales por una misma línea, en uno o en los dos sentidos.

Multiplexor *(Multiplexer)* Nombre que recibe el canal que permite que dos o más periféricos se comuniquen con la unidad central de proceso al mismo tiempo y por este mismo canal. Estos periféricos son lentos y su velocidad de manejo es mucho menor que la velocidad de transferencia del canal.

Multiproceso *(Multiprocessing)* Método de trabajo de una computadora que consiste en la ejecución simultánea de varios programas a cargo de varias unidades centrales de proceso o procesadores.

Multiprogramación *(Multiprogramming)* Método de trabajo de una computadora que consiste en la ejecución concurrente de varios programas que se encuentran en memoria al mismo tiempo. (V. PROGRAMA CONCURRENTE.)

Multipuesto *(Multipost)* Conexión de varios puestos de trabajo o terminales a una misma computadora para poder compartir los dispositivos de almacenamiento externo, la unidad central de proceso, periféricos, etcétera.

Multitarea *(Multitasking)* Se dice de una computadora que puede ejecutar varios programas concurrentemente y que no tiene necesidad de tener conectado más de un puesto de trabajo.

Multiusuario *(Multiuser)* Se dice de la computadora que trabaja con más de un usuario a través de varios puestos de trabajo conectados a ella. Normalmente, cuando las computadoras son multiusuarias, también son multitarea; sin embargo, puede haber computadoras que sean multitarea y no multiusuario (monousuario).

N

NAND *(Not-And)* Negación de la función and. Su tabla de verdad es:

	0	1
0	1	1
1	1	0

Nanosegundo *(Nanosecond)* Unidad de tiempo que equivale a 10^{-9} segundos.

NE *(Not Equal)* No igual a. Operador lógico que se utiliza en la mayoría de lenguajes.

NMOS *(Negative Metal Oxyde Semiconductor)* Circuitos MOS, pero con carga negativa. (V. MOS.)

Nodo *(Node)* Cada una de las computadoras o sistemas inteligentes que se encuentran conectados dentro de una red. Estos sistemas intercambian información y pueden utilizar los recursos propios y los ajenos.

NOR *(Not-Or)* Negación de la función OR. Su tabla es:

	0	1
0	1	0
1	0	0

Notación binaria *(Binary notation)* Expresión de un número en base dos.

Núcleo *(Kernel)* Parte del sistema operativo que permanece constantemente en la memoria principal. Siempre ocupa las mismas posiciones de memoria.

Número de secuencia *(Sequence number)* Clave numérica de control del orden de cada uno de los elementos de un conjunto.

O

Octal *(Octal)* Sistema de numeración en base 8. Los dígitos son 0, 1, 2, 3, 4, 5, 6 y 7. Este sistema se realiza con grupos de 3 bits. En computación, su uso es muy limitado; se prefiere el sistema hexadecimal.

Octeto *(Byte)* V. BYTE.

Ofimática Estudio del conjunto de tecnologías usadas en la oficina automatizada, así como procesadores de texto, ordenadores, facsímiles, correo electrónico, etc.

Operación aritmética *(Arithmetic operation)* Cualquier proceso que siga las reglas aritméticas (suma, resta, producto, etc.), efectuado con operandos numéricos, por el que se obtienen resultados expresados numéricamente.

Operación lógica *(Logical operation)* Proceso en el que se utilizan operandos lógicos (OR, AND, NOT, etc.).

Operador *(Operator)* Persona encargada de realizar las operaciones manuales que una máquina requiere. || Carácter que indica una operación aritmética o lógica; por ejemplo, el signo de sumar (+), de restar (−), de multiplicar (×), menor que <, mayor que >, la función OR, etcétera.

Operando *(Operand)* Cada uno de los elementos que en una instrucción acompañan al código de operación. Por ejemplo, en una instrucción «suma» debe haber dos operandos junto al código de operación de la suma, operandos que serán sumados para obtener un resultado.

Optimización *(Optimisation)* Modificación de las partes hardware o software para poder mejorar los resultados, ya sea en tiempo, en cantidad o en calidad de la computadora.

OR Función lógica cuya salida es 1 o verdadera si una u otra o todos los elementos de entrada son verdaderos o tienen valor 1, y 0 o falsa si todos los elementos de entrada son 0 o falsos. Su tabla de verdad es:

	0	1
0	0	1
1	1	1

OR exclusivo *(OR exclusive function)* Función lógica derivada de la función OR, pero que no se cumple cuando no

coinciden las variables de entrada; es decir, si ambas presentan el valor 1 o verdadero, o 0 o falso:

	0	1
0	0	1
1	1	0

OR inclusivo *(OR inclusive function).* V. OR.

Ordenación *(Sort)* Acción de clasificar una serie de elementos (datos, bits) según un criterio definido previamente.

Ordenador *(Computer)* V. COMPUTADORA.

Ordinograma *(Flow-chart)* Diagrama de flujo. Representación gráfica mediante señales indicativas (un rombo significa una derivación, un rectángulo una orden, etc.) de un problema determinado, para su solución. Es el paso siguiente al del algoritmo y el anterior al programa.

Organigrama *(Flow-chart)* En sentido amplio, esquema de la organización de una empresa o similar en el que se pueden observar las relaciones entre los elementos que la componen, sus funciones, la jerarquía establecida, etc. En computación, a veces se usa como sinónimo de *Ordinograma*.

Orientación *(Prompt)* Mensaje, lanzado por la computadora a través de una pantalla o un terminal, que indica al usuario que responda mediante entrada de información por el teclado.

OS *(Operating System)* V. SISTEMA OPERATIVO.

Osciloscopio *(Scope)* Aparato que se utiliza para realizar pruebas y con el que se pueden visualizar formas y señales eléctricas dependientes del tiempo.

P

Palabra *(Word)* Conjunto de bits que, como unidad elemental, puede manipular una computadora. La longitud en bits de una palabra en una computadora puede ser de 8, 16, 32, etc., y depende del microprocesador de su unidad central de proceso.

Palabra de paso *(Password)* Serie de caracteres alfanuméricos que puede reconocer un sistema con el propósito de dar acceso a datos protegidos. (También recibe el nombre de *contraseña*.)

Palabra reservada *(Keyword)* Palabras que, normalmente cuando se construyen programas, no se pueden utilizar fundamentalmente como nombres de variables.

Palanca de control *(Joystick)* V. PALANCA DE MANDO.

Palanca de mando *(Joystick)* Palanca vertical que puede inclinarse en todas direcciones con el fin de dar dirección a un movimiento. Se utiliza para desplazar un punto, el cursor o una figura por la pantalla. Es uno de los periféricos de entrada más utilizados en juegos electrónicos.

Pantalla *(Screen)* Dispositivo de *output* en el que se visualizan las informaciones que el usuario de una computadora digita en el teclado o las que la misma computadora genera.

Paquete *(Package)* Programa o juego de programas que es considerado como estándar, es decir, que puede ser útil a más de un usuario y que, sin embargo, es susceptible de modificaciones para los usuarios a los que no satisface plenamente. (Paquete de contabilidad, paquete de nóminas, etcétera.)

Parámetro *(Parameter)* Variable que puede tomar un valor diferente cada vez que se ejecuta una subrutina en la que se utiliza tal variable. Dentro de un programa, es una variable que toma un valor constante durante toda la ejecución del mismo.

Paridad *(Parity)* Sistema utilizado como medida de seguridad en las transmisiones de información o en la grabación de datos en memoria. Consiste en el uso de un bit suplementario en cada palabra de la computadora, que indica si tiene un número par o impar de bits cuyo valor sea 1.

Parpadeo *(Blinking)* Intermitencia de una palabra o de una porción de texto dentro de una pantalla, produciéndose el efecto de encendido-apagado de aquéllos.

Partición *(Partition)* Zona de la memoria principal, de una computadora que trabaja en multiprogramación, que se reserva para contener uno de los programas que se está ejecutando junto con los datos con los que opera. Al trabajar así, una computadora puede tener varias divisiones, y cada una de ellas con todos los datos especificados.

PASCAL Lenguaje de programación de alto nivel con el que se puede realizar una programación totalmente estructurada. Fue creado por Niklaus Wirth y su nombre le viene del matemático y filósofo francés Blaise Pascal.

Paso a paso *(Single step)* Ejecución de un programa instrucción a instrucción con una parada entre una y otra. Se utiliza para comprobar si los programas están bien construidos.

Paso de programa *(Program step)* Un paso de programa se corresponde con la instrucción más elemental que una computadora puede ejecutar. Por lo tanto, al ejecutar una instrucción más compleja, pueden ejecutarse varios pasos de programas.

Patilla *(Die, pin)* Terminal mecánico de conexión de un chip. Un chip se conecta con todos los demás elementos de una computadora a través de sus patillas.

P.C. *(Personal Computer)* Abreviación de *computadora personal*.

Perforación *(Punching)* Pequeño orificio efectuado sobre una cinta de papel o sobre una tarjeta, como representación de un bit de valor 1.

Perforadora de cinta de papel *(Paper-tape punch)* Aparato encargado de realizar las perforaciones en una cinta de papel. La perforación se realiza a razón de 6 caracteres por centímetro, y la velocidad de perforación varía entre 15 y 160 caracteres por segundo.

Perforadora de tarjetas *(Card punch)* Aparato encargado de realizar las perforaciones en una tarjeta. En una tarjeta o ficha pueden perforarse hasta 80 caracteres como máximo. La velocidad de perforación puede variar entre 100 y 500 fichas por minuto.

Periférico *(Peripheral)* Aparato, dispositivo o unidad que no forma parte de la unidad central de una computadora, pero que, conectado con ésta, sirve para almacenar información o como dispositivo de entrada/salida de datos. Son periféricos las pantallas, las unidades de disco, las perforadoras, las impresoras, etcétera.

Pico *(Pico)* Prefijo que significa 10^{-12}.

Picosegundo *(Picosecond)* Unidad de medida de tiempo, equivalente a 10^{-12} segundos.

PILOT Lenguaje de programación concebido para la enseñanza. Su utilización no requiere ningún conocimiento de computación y permite, por lo tanto, a una persona sin estos conocimientos la construcción de programas de enseñanza bastante complejos.

Pista *(Track)* Cada una de las bandas capaces de recibir información en un dispositivo magnético. Se trata de bandas longitudinales en cintas magnéticas, o circunferencias concéntricas en un disco, o circunferencias coaxiales en un tambor magnético.

Pixel *(Picture element)* Elemento más pequeño de una pantalla al que se le puede dar intensidad y color. Equivale al significado de un punto. Cuantos más pixels haya en la pantalla de una computadora, más resolución poseerá ésta.

PL/1 *(Programming language 1)* Lenguaje de programación 1. Lenguaje de programación de alto nivel que se creó como alternativa a los lenguajes ALGOL, FORTRAN y COBOL. Se utiliza normalmente en computadoras de una gran potencia, aunque existe también una versión para microcomputadoras.

Placa *(Board)* Soporte donde se montan los circuitos electrónicos. En las placas de una computadora se encuentran los chips de memoria, los microprocesadores, etc. Están constituidas de vidrio o baquelita.

PL/M *(Programming Language Microprocessor)* Lenguaje de programación para microprocesadores de la compañía INTEL. Es un lenguaje de alto nivel.

Plotter Periférico gráfico de salida que permite realizar trazos gráficos al ser controlado por una computadora. Puede dibujar cualquier tipo de gráfico por complejo que sea.

Portabilidad *(Portability)* Propiedad de un software para ser utilizado en varias computadoras.

Posición de memoria *(Memory location)* Cada una de las unidades elementales que pueden contener información en la memoria central. Estas posiciones se definen mediante direcciones.

Potencia *(Power)* Energía eléctrica que alimenta a una computadora y, en general, a un equipo eléctrico o electrónico.

Preparado *(Ready)* Dícese de cualquier periférico de una computadora cuando se encuentra en condiciones de establecer diálogo con dicha computadora.

Prioridad *(Priority)* Orden de prelación en la ejecución de un proceso. Cuando en una computadora deben ejecutarse varios procesos y no todos pueden ejecutarse a la vez, según la prioridad de cada uno, se ejecutarán antes o después.

Procedimiento *(Procedure)* Serie de pasos que deben seguirse para resolver un problema. A veces se le llama procedimiento a un programa o subprograma.

Procesador *(Processor)* Dispositivo electrónico que controla las operaciones que deben efectuarse en una computadora para obtener los resultados apetecidos. Forma la unidad central de proceso. Una computadora puede tener varias CPU, compuestas a su vez por varios procesadores. || Programa del sistema que realiza las traducciones, por ejemplo, un compilador.

Procesamiento *(Processing)* V. TRATAMIENTO.

Procesar *(Process)* Acción de tratar la información o los datos. (V. TRATAMIENTO.)

Programa *(Program)* Conjunto de instrucciones secuenciales, correspondientes a un algoritmo escrito en cualquier lenguaje de programación, con las que se puede realizar un trabajo determinado mediante la ejecución de tales instrucciones por parte de la computadora.

Programa almacenado *(Stored program)* Conjunto de instrucciones residentes en la memoria externa y que en cualquier momento pueden reclamarse o leerse para ser ejecutadas por la computadora.

Programa cargador *(Loader)* V. CARGADOR.

Programa compilador *(Compiler)* V. COMPILADOR.

Programa concurrente *(Concurrent program)* Programa dividido en secciones que permiten su ejecución simultánea. Esta simultaneidad debe manifestarse en el programa a través de sentencias indicativas. El programa concurrente permite la multiprogramación.

Programa cruzado *(Cross program)* Programa de una computadora que es ejecutado por otra. Esto es posible gracias a un ensamblador o compilador cruzado, que traduce el lenguaje que funciona en la primera computadora al código que la segunda puede utilizar.

Programa de prueba *(Benchmark)* Programa normalizado que puede emplearse en distintas computadoras para ensayar sus características de volocidad, eficacia y precisión, y compararlas.

Programa de servicio *(Utility program)* Programa que no forma parte del sistema operativo, pero que sí forma parte del software de sistema y que generalmente permite al usuario realizar tareas habituales (copia de ficheros, formatación de discos o cintas, etcétera).

Programa de usuario *(User program)* Programa creado por el usuario para resolver los problemas individuales del mismo.

Programa ensamblador *(Assembly program)* Programa que permite convertir en lenguaje máquina un programa escrito en un lenguaje de bajo nivel o ensamblador. (V. ENSAMBLADOR.)

Programa fuente *(Source program)* Programa escrito en lenguaje fuente, por lo que la máquina no puede ejecutarlo hasta que el compilador no genere el programa objeto.

Programa objeto *(Object program)* Programa en el lenguaje objeto, que es el resultado de la compilación de un programa fuente. Este programa objeto puede ser o bien procesado directamente por la computadora, o bien puede sufrir un proceso que lo transforme a continuación en lenguaje máquina ejecutable.

Programa principal *(Main program)* Programa que lleva el control del proceso y que lo transfiere a otra parte del programa llamada subprograma.

Programa secuencial *(Sequential program)* Programa diseñado de forma que sus instrucciones siguen un orden lineal riguroso.

Programación *(Programming)* Técnica de confección de programas.

Programación estructurada *(Structured programming)* Método de programación que pretende evitar al máximo los errores en los programas. Su técnica es la de dividir el programa en bloques o etapas que se irán definiendo y concretando en forma descendente hasta llegar a completar el programa o resolver el problema.

Programación lógica *(Logic programming)* Tipo de programación en la cual las sentencias son sentencias lógicas, tales como implicaciones lógicas, proposiciones lógicas, etcétera.

Programador *(Programmer)* Persona versada en técnica de programación y que confecciona programas para computadoras.

Programador de aplicaciones (Applications programmer) Persona que diseña programas para que el usuario resuelva problemas específicos. Tal conjunto de programas se denomina *aplicaciones*.

Programador de sistemas *(System programmer)* Persona encargada de realizar programas de utilidad o de realizar rutinas de ayuda al sistema operativo o incluso de modificar rutinas de este tipo.

PROLOG Término formado con las primeras sílabas de las voces francesas *Programation Logique*. Lenguaje de programación cuyas instrucciones representan fases lógicas. Este lenguaje presenta una estructura totalmente distinta a la de los lenguajes simbólicos de alto nivel. Se usa como el LIST en inteligencia artificial.

PROM *(Programmable Read Only Memory)* V. MEMORIA PROGRAMABLE DE SÓLO LECTURA.

Protección contra grabación *(Write-protect)* Dispositivo físico que se adhiere o coloca (o ya está presente) en la funda o chasis de un soporte de memoria de almacenamiento externo (diskette, cinta, etc.) y que puede ser detectado por la unidad en la que se introduce, con el fin de que tal soporte no pueda sufrir ninguna grabación.

Protección contra lectura *(Read protect)* Método que impide que personas no autorizadas tengan acceso a determinados archivos o ficheros.

Protocolo *(Protocol)* Conjunto de reglas y métodos que hay que seguir para intercambiar información entre dos computadoras, conectadas normalmente por comunicación telefónica.

Pseudoinstrucción *(Pseudo-instruction)* Instrucción o sentencia perteneciente a un lenguaje de compilación que se mezcla con las otras instrucciones básicas que dicho lenguaje debe compilar.

Puesta en marcha *(Power on, power up)* Acción de poner en funcionamiento cualquier sistema electrónico en general y de una computadora en particular.

Puesto de trabajo *(Work station)* Cualquier terminal compuesto como mínimo por una pantalla y un teclado, conectado a una computadora y desde el que se puede realizar cualquier trabajo de programación, consulta, etcétera.

Pulgada *(Inch)* Unidad inglesa de longitud que equivale a 2,54 cm. Se utiliza para medir la longitud de los módulos de papel en la impresora.

Punto de entrada *(Entry point)* Punto o instrucción concreta a partir de la cual se reanuda un programa después de haberse producido una interrupción.

Punto de interrupción *(Breakpoint)* Dirección de memoria, especificada por el usuario, que pertenece a un programa y en la que la ejecución del mismo se detendrá para poder determinar los valores de las variables y de algunos estados de la computadora.

Punto de una imagen *(Pixel)* V. PIXEL.

Q

QWERTY Nombre que recibe el teclado de una computadora en el que la disposición de las teclas de la primera fila siguen de derecha a izquierda la misma disposición que el nombre QWERTY.

R

RAM *(Random Access Memory)* Memoria de acceso directo o aleatorio. (V. MEMORIA DE ACCESO DIRECTO.)

Ranura *(Slot)* Cada uno de los zócalos que puede tener una computadora en los cuales pueden introducirse placas o tarjetas impresas.

Rebote *(Bouncing)* Vibraciones que se producen al abrir o cerrar un interruptor mecánico. Para eliminar estas vibraciones o ruidos molestos se usa un dispositivo de hardward o de software llamado *debouncing*.

Recogida de datos *(Data collection)* Acción de reunir la información que se va a procesar en un punto central del sistema.

Red *(Network)* Conjunto de nodos conectados entre sí. Pueden formar redes locales, privadas o públicas. V. NODO.

Red ARPA *(Advanced Research Projects Agency Network)* Red de agencia de proyectos de investigación avanzada. Red de computadoras del departamento de Defensa de Estados Unidos de América que abarca un área geográfica muy extensa.

Red Ethernet Red de computadoras en la que no hay nodo de control y en la que los nodos se van colocando a lo largo de un cable transmisor.

Registrar *(Record)* Almacenar datos digitales.

Registro lógico *(Record)* Conjunto de uno o más campos que forman un grupo de datos similares relacionados entre sí. Una colección de registros es un fichero.

Reloj *(Clock)* V. RELOJ INTERNO.

Reloj interno *(Internal timer)* Reloj que se encuentra en el interior de una computadora y que, a la vez que proporciona la hora como un reloj digital cualquiera, sincroniza todas las operaciones que la computadora efectúa.

Repertorio de instrucciones *(Instruction repertoire)* Conjunto de instrucciones que forman parte de un lenguaje de programación.

Retardo *(Delay)* Desfase de tiempo que se da a una señal respecto a otra tratada simultáneamente en los circuitos.

Retorno de carro *(Carriage return)* Tecla que se encuentra en todos los teclados de cualquier computadora y que se utiliza para finalizar cualquier entrada de datos. Al ser pulsada, el cursor se desplaza de una línea a la siguiente.

Retroceso *(Backspace)* Operación, normalmente efectuada con una tecla del teclado, que consiste en hacer retroceder el cursor un espacio en la pantalla.

Reubicable *(Relocatable)* Dícese del programa que puede cargarse en cualquier lugar de la memoria para después ser ejecutado. Para ser reubicable, dicho programa no puede contener instrucciones con direccionamientos absolutos.

Robótica Estudio de la construcción, ensamblaje, generación, programación y uso de los robots y autómatas en general.

ROM *(Read Only Memory)* V. MEMORIA DE SÓLO LECTURA.

Rótulo *(Label)* Carácter o conjunto de caracteres que se utilizan para identificar una sentencia, un dato, un conjunto de sentencias o un conjunto de datos.

RPG *(Report Program Generator)* Generador de programas de informes. Lenguaje de programación usado preferentemente en gestión de empresas.

RTL *(Register Transfer Language)* Lenguaje de transferencia entre registros. Lenguaje que describe el funcionamiento físico (hardward) de la computadora y que está basado en operaciones entre registros internos.

RTOS *(Real Time Operating System)* Sistema de explotación en tiempo real. (V. SISTEMA OPERATIVO EN TIEMPO REAL.)

Ruido *(Noise)* Cualquier perturbación eléctrica o electromagnética que pueda producir interferencias en las transmisiones de información vía telefónica.

Rutina *(Routine)* Pequeño programa que nunca se ejecuta como programa individual, aunque podría hacerse, y que siempre forma parte de otro programa principal desde el cual se accede al anterior.

Rutina de diagnóstico *(Diagnostic routine)* Rutina que se utiliza para hacer trabajar independiente y sucesivamente todos los elementos de un equipo con el fin de detectar un posible fallo.

R/W *(Read-write)* Lectura/escritura.

S

Salida *(Output)* Cualquier información que se obtiene de una computadora, ya sea a través de pantalla o a través de la impresora, o incluso aquella que puede obtenerse grabada en un dispositivo de almacenamiento externo.

Salto *(Skip)* Se denomina así a un cierto tipo de instrucción existente en algunos lenguajes que sirve para interrumpir el proceso secuencial de ejecución de un programa para realizar un salto de una o más instrucciones.

Salto condicional *(Conditional jump)* Salto en la ejecución de un programa, pero que se producirá sólo si se cumple una determinada condición.

Salto de papel *(Paper throw)* El avance normal de una impresora cuando está imprimiendo es de línea en línea. Sin embargo, a veces el programa que se está ejecutando hace que la impresora avance varias líneas hasta colocarse en una específica. Este avance se denomina *salto de papel*.

Sector *(Sector)* Cada una de las zonas en que se dividen físicamente las pistas de un disco o tambor. Un sector puede contener físicamente parte de un registro lógico, uno o varios. Además, es la unidad física más pequeña de almacenamiento o registro físico.

Secuencial *(Sequential)* V. ACCESO.

Segmento *(Segment)* Parte de un programa que se carga de una sola vez en memoria.

Seguridad de los datos *(Data security)* Cualquier método utilizado para proteger los datos almacenados en los dispositivos de almacenamiento externo contra el acceso a ellos de personas no autorizadas.

Semántica *(Semantics)* Parte de la lingüística que estudia la relación entre un símbolo y su significado. En computación, estudia la relación entre los símbolos utilizados en las instrucciones y su significado.

Sentencia *(Sentence)* V. INSTRUCCIÓN.

Seudoinstrucción *(Pseudo-instruction)* V. PSEUDOINSTRUCCIÓN.

Silicio *(Silicon)* Elemento químico con el que actualmente se fabrican la mayoría de los circuitos integrados.

Símbolo *(Symbol)* Carácter o conjunto de caracteres que sirve para representar una cantidad, operación, información, instrucción, etcétera.

Simulación *(Simulation)* Representación del funcionamiento de un determinado proceso por medio de la computadora.

Síncrono *(Synchronous)* Proceso que para su óptima ejecución depende de una señal de reloj, es decir, del tiempo.

Sistema *(System)* Conjunto de elementos interdependientes. Por ejemplo, una computadora es un sistema formado por todos sus elementos, que son interdependientes. También significa el conjunto de axiomas y reglas que hacen que se desarrolle perfectamente un determinado sistema.

Sistema de desarrollo *(Development system)* Computadora que se utiliza para desarrollar el software o hardware de otros sistemas.

Sistema de gestión de ficheros *(File management system)* Programa que, formando parte de un sistema operativo, permite el manejo de ficheros de una forma transparente, es decir, sin tener en cuenta el tipo de la unidad en la que éstos se encuentran.

Sistema de tiempo compartido *(Time sharing system)* Sistema en el que el tiempo de ejecución de la unidad central de proceso es compartido entre varios programas para poder realizar varias tareas de forma concurrente.

Sistema dedicado *(Dedicated system)* Computadora que sólo puede ejecutar las tareas para las cuales ha sido concebida.

Sistema operativo *(Operating system)* Programa o conjunto de programas que sirven para controlar todas las operaciones que puede efectuar una computadora, como por ejemplo asignación de memoria para los procesos en ejecución, asignación de los recursos de entrada/salida, etc.

Sistema operativo en tiempo real *(Real time operating system)* Sistema operativo de una computadora que hace que cualquier consulta o demanda de datos por parte de uno o varios usuarios sea contestada de forma inmediata.

Software Conjunto de programas que puede ejecutar una computadora. El software se divide en dos clases: *software del sistema* y *software aplicativo* o de aplicación.

Software de aplicación *(Application software)* Conjunto de programas escritos en cualquier lenguaje de programación que sirven para resolver, mediante la computadora, los problemas de una aplicación determinada.

Software del sistema *(System software)* Conjunto de programas que hacen que la computadora pueda ejecutar un programa escrito por el usuario. Entre ellos se encuentra el sistema operativo, los lenguajes, el ensamblador, utilizados para poder copiar ficheros o borrarlos, etcétera.

Soporte *(Support)* Material normalmente destinado a recibir información y mantenerla de forma que pueda ser leída por la computadora, como cintas, discos, etcétera.

Subíndice *(Subscript)* Variable que según el valor que tome sirve para designar un elemento u otro de una lista o matriz de elementos.

Subprograma *(Subprogram)* Conjunto de instrucciones que se utiliza entre varios puntos de un programa o que puede insertarse en diferentes programas.

Subrutina *(Subroutine)* V. SUBPROGRAMA.

Supervisor *(Supervisor)* Programa que forma parte del sistema operativo de una computadora. (V. MONITOR.)

T

Tabla de datos *(Table)* Lista o conjunto de listas en las que se especifican los datos de una determinada información.

Tabla de verdad *(Truth table)* Tabla en la que se especifican los valores de salida de una función u operación al aplicar ésta a dos valores de entrada.

Tablero digitalizador *(Digitizer table)* V. DIGITALIZADOR.

Tabulación *(Tab)* Acción conseguida al tabular.

Tabular *(Tab)* Desplazar el cursor a una posición determinada, saltando varias posiciones de pantalla, para visualizar información a partir de esa posición. Este desplazamiento se consigue a través de una sentencia de programa o mediante la pulsación de una tecla del teclado.

Tamaño de instrucción *(Instruction lenght)* Número de bytes de que consta una instrucción.

Tamaño de la pantalla *(Screen size)* Dimensiones físicas de la pantalla expresadas en pulgadas. Para efectuar la medida se toma la longitud de la diagonal de la pantalla, expresada en pulgadas. A veces se mide también en caracteres. Así, cuando decimos que el tamaño de una pantalla es de 30×25 queremos expresar que se pueden visualizar hasta 25 líneas de 80 caracteres cada una.

Tamaño de memoria *(Memory size)* Número de caracteres o bytes que pueden estar almacenados en una memoria. El tamaño de una memoria se expresa generalmente en Kbytes. Sabiendo que un Kbyte equivale a 1.024 bytes, si el tamaño de memoria de una computadora es de 256 Kbytes, ésta podrá almacenar 256×1.024 bytes.

Tarea *(Task)* Trabajo que tiene que desarrollar un equipo o sistema a partir de una determinada planificación con el fin de obtener unos resultados determinados. En computación significa cada uno de los trabajos que puede ejecutar una computadora de una manera concurrente.

Tarjeta de control *(Control card)* Tarjeta perforada que contiene información especial. Son tarjetas de control las tarjetas maestras y las de identificación.

Tarjeta magnética *(Magnetic card)* Tarjeta que tiene adherido un material magnético gracias al cual se puede almacenar información en ella. Esta información puede ser leída o grabada por la computadora; en el primer caso los datos serán de entrada y en el segundo de salida.

Tarjeta perforada *(Punched card)* Soporte, normalmente de cartón, que sirve como entrada de datos a una computadora. Cada carácter se representa mediante una combinación de perforaciones en la tarjeta. En ésta pueden representarse hasta ochenta caracteres. Actualmente ya no se usan las tarjetas perforadas como entrada de datos.

Tarjeta perforada de identificación *(Identification card)* Tarjeta perforada de control que precede a las tarjetas de programa y que contiene información general sobre este programa.

Tarjeta perforada maestra *(Master card)* Tarjeta de control que se coloca al principio o al final de un lote de tarjetas y que normalmente contiene instrucciones para el procesador.

Tecla *(Key)* Cada uno de los elementos de que consta un teclado. El teclado se encuentra dividido en grupos de teclas, cada uno de los cuales tiene una función diferente.

Tecla de carácter *(Character key)* Tecla cuya función es entrar textos alfanuméricos en la memoria.

Tecla de control *(Control key)* Tecla cuya función es ordenar al sistema operativo que se realice una determinada acción.

Tecla de función programable *(Soft key)* Tecla cuya función puede variarse a voluntad del usuario y mediante un programa para que al pulsarla se visualice una determinada información en la pantalla.

Tecla de movimiento del cursor *(Arrow key)* Cualquiera de las cuatro teclas que se usan para mover el cursor por la pantalla de una computadora o de un terminal en las cuatro direcciones posibles: arriba, abajo, derecha e izquierda.

Teclado *(Keyboard)* Conjunto de teclas utilizadas para introducir información en la computadora. En la mayoría de computadoras el teclado es un periférico que se encuentra separado de la unidad de base. En un terminal, el teclado también está separado de la pantalla.

Teclado numérico *(Numeric keypad)* Conjunto de teclas situadas generalmente en la parte derecha del teclado de una computadora o terminal. Se trata de teclas numéricas que sirven para entrar datos numéricos en la memoria de una computadora. La disposición de este grupo de teclas facilita la entrada repetitiva de datos numéricos.

Telecomunicaciones *(Telecomunications)* Nombre genérico que se aplica a la transmisión de datos a larga distancia.

Telemática Integración de las telecomunicaciones con el cálculo automático o proceso de datos, produciendo nuevas aplicaciones y servicios para el tratamiento y distribución de la información entre usuarios muy alejados.

Teleproceso *(Teleprocessing)* Tratamiento de la información cuyos datos son transmitidos a distancia. Los terminales se encuentran generalmente a gran distancia de la computadora, y las órdenes o la demanda de información, así como la oportuna contestación a tales órdenes o demandas por parte de la computadora, se transmiten vía telefónica.

Teletipo *(Teletype)* Terminal dedicado a comunicaciones. Consta de un teclado y una impresora. Se utiliza para enviar o recibir todo tipo de información alfanumérica. Actualmente, los teletipos se conectan a computadoras para realizar esta misma función.

Terminal *(Terminal)* Sistema compuesto por un teclado, una pantalla, un circuito de control y, algunas veces, un módem, que va conectado a una computadora y se utiliza para introducir datos y extraer resultados de la computadora. Si se usa un módem, el terminal se encuentra a larga distancia.

Terminal de pantalla *(CRT terminal)* Terminal de computadora cuya pantalla de visualización está formada por un tubo de rayos catódicos.

Terminal inteligente *(Intelligent terminal)* Terminal que puede trabajar de dos modos distintos: o bien conectado a la computadora de la que es un terminal y, por lo tanto, gobernado por ella, o bien sin conectarse a la computadora, es decir con cierta capacidad de autonomía gracias a los microprocesadores de que dispone.

Terminal no inteligente *(Dumb terminal)* Terminal de computadora que al ser desconectado de ésta queda inutilizado; no puede trabajar autónomamente.

Terminal remoto *(Remote terminal)* Terminal de computadora que se encuentra conectado con ella a larga distancia. Normalmente se necesita un módem para realizar esta conexión.

Texto alfabético *(Alphabetic text)* Conjunto de caracteres alfabéticos.

Texto alfanumérico *(Alphanumeric text)* Conjunto de caracteres alfanuméricos.

TI *(Texas Instruments)* Uno de los mayores fabricantes de componentes electrónicos y semiconductores del mundo, con sede en Houston, Texas (USA).

Tiempo compartido *(Time sharing)* V. SISTEMA DE TIEMPO COMPARTIDO.

Tiempo de acceso *(Access time)* Espacio de tiempo que transcurre entre el principio y el fin de la búsqueda o introducción de una información en una memoria (externa, de almacenamiento externo, etcétera).

Tiempo de búsqueda *(Seek time)* Tiempo que transcurre entre la recepción de la orden de procesamiento de la cabeza de lectura/grabación de un disco en una pista determinada y el posicionamiento.

Tiempo de ejecución *(Run time)* Tiempo transcurrido entre la orden de ejecución de un programa y el fin de la ejecución de éste.

Tiempo de respuesta *(Response time)* Tiempo que transcurre entre la petición de información de un terminal a la computadora y la recepción de esta información en dicho terminal.

Tiempo real *(Real time)* V. SISTEMA OPERATIVO EN TIEMPO REAL.

TOD *(Time Of Day)* Hora del día. (V. RELOJ INTERNO.)

Traductor *(Translator)* Cualquier programa compilador o ensamblador.

Transferencia *(Transfer)* Movimiento de datos de un soporte a otro, desde una posición a otra dentro del mismo soporte o de una posición de memoria a otra.

Transmisión *(Transmission)* Acto por el que se envían datos desde un sistema y se reciben estos mismos datos por otro sistema (terminales, computadoras, impresoras, etc.)

Transmisión asíncrona de datos *(Asynchronous data transmission)* Envío y recepción de datos efectuados por procedimientos asíncronos. (V. ASÍNCRONO.)

Transmisión en paralelo *(Parallel transmission)* Envío y recepción de información a través de líneas compuestas de varias vías paralelas, de tal forma que todos los bits de una palabra o carácter circulan simultáneamente.

Transmisión en serie *(Serial transmission)* Envío y recepción de datos a través de una vía única, de tal manera que los bits de una palabra o carácter circulan uno tras otro.

Transmisión síncrona de datos *(Synchronous data transmission)* Envío y recepción de datos efectuados por procedimientos síncronos. (V. SÍNCRONO).

Transportabilidad *(Portability)* V. PORTABILIDAD.

Tratamiento *(Processing)* Ejecución automática de una serie de operaciones sobre una información facilitada. Acción de operar sobre una información o unos datos desorganizados con objeto de obtenerlos organizados según criterios establecidos de antemano.

Tratamiento de textos *(Word processing)* Programa que permite entrar un texto en la computadora; al mismo tiempo o posteriormente se pueden suprimir, añadir, reemplazar, modificar, desplazar, repartir, etc., signos, palabras,

frases o párrafos enteros. El texto, una vez modificado, puede listarse en una impresora o grabarse en un dispositivo de almacenamiento externo de memoria para su posterior utilización.

Truncar *(Truncate)* En operaciones matemáticas realizadas en computadoras, despreciar las cifras menos significativas.

U

UCSD Pascal *(UCSD Pascal)* Universidad de California, San Diego. Sistema operativo del cual es propietaria la citada universidad y cuya base es el lenguaje *Pascal*.

Unidad *(Drive)* Conjunto mecánico electrónico que sirve de periférico a una computadora y que, mediante la introducción en su interior de un soporte (generalmente magnético) tal como diskettes, discos, cintas, etc., puede almacenar o suministrar información de una computadora.

Unidad central *(Central unit)* Dispositivo que conforma el núcleo de un equipo y que, al tener unidad de control, unidad lógica, memoria rápida, etc., es capaz de gestionar un conjunto de periféricos.

Unidad Central de Proceso (C.P.U.) *(Central Processing Unit)* Parte principal de una computadora. Se trata del chip que realiza todas las operaciones lógicas y cálculos numéricos fundamentales; interpreta las instrucciones dadas a la computadora y hace que ésta las ejecute.

Unidad de cartucho de cinta *(Cartridge drive)* Conjunto mecánico electrónico donde se introduce el cartucho de cinta y con el que se podrá leer y grabar.

Unidad de cassette *(Cassette drive)* Conjunto mecánico electrónico donde se introduce el cassette y con el que podrá leerse y grabarse tal soporte.

Unidad de control *(Control unit)* Unidad de un periférico que sirve para efectuar todas las operaciones propias de un periférico cuando éstas le son ordenadas por la unidad central.

Unidad de cinta magnética *(Magnetic tape unit)* Conjunto mecánico electrónico donde se introduce la cinta magnética y con el que se podrá leer y grabar.

Unidad de disco *(Disk drive)* Conjunto mecánico electrónico donde se introduce, o en muchos casos ya va incluido, un disco; con dicho conjunto se podrá leer y grabar este soporte.

Unidad de diskette *(Diskette drive)* Conjunto mecánico electrónico donde se introduce el diskette y con el que se podrá leer y grabar.

Unidad lógica *(Logic unit)* Parte de la unidad central de proceso que contiene los elementos necesarios para la realización de las operaciones lógicas.

UNIVAC 1 Computadora construida en el año 1950 con tubos de vacío. Era tan inmensa que ocupaba toda una sala.

UNIX Sistema operativo para mini y microcomputadoras creado por los laboratorios Bell. Es un sistema multiusuario y con estructura de ficheros jerarquizada.

Usuario *(User)* Persona que utiliza la computadora o persona para la cual es utilizada.

Utilidad *(Utility)* V. PROGRAMA DE SERVICIO.

V

Valor absoluto *(Absolute value)* Valor de una cantidad sin especificar su signo. Por ejemplo, el valor absoluto de +31 es 31 y el de −31 también es 31.

Valor inmediato *(Inmediate value)* Valor que se especifica en una instrucción y que la unidad central de proceso almacenará en memoria tal como se encuentra en la instrucción.

Variable *(Variable)* Dato de un proceso que puede tomar valores diferentes dentro del mismo proceso o en otras ejecuciones del mismo.

Vector *(Vector)* Conjunto de elementos (variables) que se pueden identificar individualmente utilizando un índice. Así, por ejemplo, si decimos A(5), nos estamos refiriendo al quinto elemento del vector A.

Velocidad *(Rate, Speed)* Cuando se quieren comparar las potencias de dos computadoras diferentes, normalmente uno de los factores que se tienen en cuenta es la velocidad para efectuar algunas tareas, tales como la velocidad de ejecución de su unidad central, que se mide en instrucciones por segundo, la velocidad de transferencia de datos en memoria, que se mide en caracteres por segundo, etcétera.

Velocidad de transferencia de datos *(Data transfer rate)* Velocidad de transmisión de datos entre dos puntos que pueden formar parte de la computadora (memoria, discos, cintas, etc.) o entre dos computadoras.

Ventana *(Window)* Cada una de las partes en las que puede dividirse la pantalla de una computadora. Una pantalla puede dividirse en varias ventanas, en cada una de las cuales puede visualizarse información relativa a diferentes tareas efectuadas por la computadora.

Ventana de texto *(Text window)* Ventana de la pantalla donde sólo se visualiza un texto alfanumérico. (V. VENTANA.)

Verificación *(Verification, check)* Control de la validez de unos datos. Puede efectuarse una verificación de los datos recibidos por vía telefónica desde otra computadora.

Vía *(Bus)* Camino por el que las diferentes unidades que forman la computadora se intercambian información.

Vídeo *(Video)* Aplicado a las computadoras, pantalla de un terminal o de una computadora.

Visualizar *(Display)* Hacer aparecer información en la pantalla de la computadora o del terminal.

Volumen *(Volumen)* Conjunto total de la información almacenada en un soporte de almacenamiento externo de memoria.

W

Winchester *(Winchester)* Tipo de disco magnético de gran capacidad y de pequeño formato. Su nombre viene del prototipo IBM 3030, cuyo número se asoció con el rifle Winchester de calibre 30-30.

Word Star *(Word Star)* Paquete de tratamiento de textos muy difundido actualmente. Fue desarrollado por la compañía americana Micro Pro (Micro Pro International Corp., San Rafael, California, USA).

X

XENIX Versión del sistema operativo UNIX, desarrollada por la compañía Microsoft.

XON-XOFF Protocolo de transmisión serial de datos entre una computadora y una impresora. XON es el nombre de un carácter ASCII cuyo código envía la impresora a la computadora para indicarle que está lista para recibir. XOFF es el nombre de otro carácter ASCII cuyo código envía la impresora a la computadora para indicarle que no envíe más información a causa de que el buffer de impresión está ya lleno. Esto se produce porque la computadora es más veloz en transmitir datos que la impresora en imprimirlos.

Z

Zona *(Zone)* Parte de la memoria central que puede reservarse para una función determinada.

Vocabulario inglés-español

A

ABEND	ABEND
Abort	Abort
Absolute value	Valor absoluto
ACC	ACC
Access	Acceso
Access method	Método de acceso
Access time	Tiempo de acceso
Accumulator	Acumulador
ACIA	ACIA
ACK	ACK
Ada	Ada
Address	Dirección
AI	AI
ALGOL	ALGOL
Algorithm	Algoritmo
Alphabet	Alfabeto
Alphabetic	Alfabético
Alphabetic text	Texto alfabético
Alphanumeric	Alfanumérico
Alphanumeric text	Texto alfanumérico
Alterable memory	Memoria alterable
Amplifier	Amplificador
Analyst	Analista
Analog data	Datos analógicos
AND	AND
ANSI	ANSI
APL	APL
Application	Aplicación
Application programmer	Programador de aplicaciones
Application software	Software de aplicación
Archive	Archivo
Area	Área
Argument	Argumento
Arithmetic operation	Operación aritmética
ARPA	ARPA
Array	Matriz
Arrow key	Tecla de movimiento del cursor
Artificial intelligence	Inteligencia artificial
ASCII	ASCII
Assembler	Ensamblador
Assembler language	Lenguaje ensamblador
Assembly listing	Listado ensamblado
Assembly program	Programa ensamblador
Assignement	Asignación
Asynchronous	Asíncrono
Asynchronous data transmission	Transmisión asíncrona de datos
Automata	Autómata
Automation	Automatización
Auxiliary memory	Memoria auxiliar

B

Backspace	Retroceso
Bar code	Código de barras
Bar code reader	Lector de código de barras
Bar printer	Impresora de barras
BASIC	BASIC
Baud	Baudio
Benchmark	Programa de prueba
Bidirectional	Bidireccional
Bidirectional printer	Impresora bidireccional
Binary	Binario
Binary language	Lenguaje binario
Binary notation	Notación binaria
Bit	Bit
Blank	Blanco
Blinking	Parpadeo
Block	Bloque
Board	Placa, tarjeta
BOF	BOF
Boolean logic	Lógica booleana
Bouncing	Rebote
BPI	BPI
Bubble memory	Memoria de burbujas
Buffer	Buffer
Bump	Memoria anexa
Bus	Vía
Byte	Byte, octeto

C

C	C
CAD	CAD
CAD/CAM	CAD/CAM
CAE	CAE
Call	Llamada
CAM	CAM
Card	Ficha
Card punch	Perforadora de tarjetas
Card reader	Lector de tarjetas
Cartridge	Cartucho
Cartridge drive	Unidad de cartucho de cinta
Cassette	Cassette
Cassette drive	Unidad de cassette
CBASIC	CBASIC
Central memory	Memoria central
Central Processing Unit	Unidad central de proceso
Central unit	Unidad central
Clear	Borrar, limpiar
CLK	CLK
Clock	Reloj
Close	Cerrar
CMOS	CMOS
COBOL	COBOL
Code	Código
Command	Comando, orden
Compatibility	Compatibilidad
Compile	Compilar
Compiled	Compilado
Compiler	Compilador, programa compilador
Compiler language	Lenguaje compilador
Computer	Computadora, ordenador
Computing	Computación, informática
Concurrent program	Programa concurrente
Concurrent programming language	Lenguaje de programación concurrente
Conditional call	Llamada condicional
Conditional jump	Salto condicional
Configuration	Configuración
Console	Consola
Constant	Constante
Contents addressable memory	Memoria asociativa
Control card	Tarjeta de control
Control key	Tecla de control
Control unit	Unidad de control
Conversational	Conversacional
Conversational language	Lenguaje conversacional
Converter	Convertidor
Counter	Contador
CPI	CPI
CP/M	CP/M
CP/M-86	CP/M-86

CPS	CPS
CPU	CPU
Cross program	Programa cruzado
CRT	CRT
CRT terminal	Terminal de pantalla
Cursor	Cursor
Cybernetics	Cibernética

CH

Chain printer	Impresora de cadena
Chaining	Encadenamiento
Channel	Canal
Character	Carácter
Character key	Tecla de carácter
Character set	Juego de caracteres
Check	Verificación
Chip	Chip

D

DAC	DAC
Daisy wheel printer	Impresora de margarita
Data	Dato
Data bank	Banco de datos
Data base	Base de datos
Data collection	Recogida de datos
Data security	Seguridad de los datos
Data transfer rate	Velocidad de transferencia de datos
DBMS	DBMS
Debugger	Debugger
Decimal	Decimal
Decode	Decodificar
Decoder	Decodificador
Decrement	Decremento
Dedicated system	Sistema dedicado
Delay	Retardo
Density	Densidad
Development system	Sistema de desarrollo
Diagnostic routine	Rutina de diagnóstico
Die	Patilla
Digit	Cifra, dígito
Digital data	Datos digitales
Digital Research	Digital Research
Digitize	Digitalizar
Digitizer	Digitalizador
Digitizer table	Tablero digitalizador
Directive	Directiva
Directory	Directorio
Disk	Disco
Disk drive	Unidad de disco
Disk memory	Memoria de disco
Diskette	Diskette
Diskette drive	Unidad de diskette
Display	Display, visualizar
DMA	DMA
Drive	Unidad
Dump terminal	Terminal no inteligente
Dynamic memory	Memoria dinámica

E

EBCDIC	EBCDIC
Edit	Editar
Editor	Editor
Emulator	Emulador
Encode	Codificar
ENIAC	ENIAC
Entry point	Punto de entrada
EQ	EQ
Error message	Mensaje de error
Ethernet	Ethernet
Execute	Ejecutar

F

FD	FD
Fetch	Lectura de instrucción
Field	Campo
FIFO	FIFO
File	Fichero
File management system	Sistema de gestión de ficheros
Floppy disk	Diskette
Flow-chart	Ordinograma, organigrama
Format	Formatar, formato
Formater	Formatador
FORTH	FORTH
FORTRAN	FORTRAN
Full duplex	Full duplex

G

Generate	Generar
Generating	Generador
Gigabyte	Gigabyte, gigaocteto
GT	GT

H

Handler	Manipulador
Hard copy	Hard copy
Hard disk	Disco duro
Hardware	Hardware
Head	Cabezal
Hexadecimal	Hexadecimal
High-level language	Lenguaje de alto nivel
Hollerith code	Código Hollerith
Home computer	Computadora doméstica
HP	HP
HZ	HZ

I

IBM	IBM
Identification card	Tarjeta perforada de identificación
Inch	Pulgada
Increment	Incremento
Index	Índice
Indexed	Indexado
Indexing	Indexación
Indicator	Indicador
Indirect	Indirecto
Initialization	Inicialización
Ink jet printer	Impresora de chorro de tinta
Inmediate value	Valor inmediato
Input	Entrada
Insert	Insertar
Instruction	Instrucción
Instruction decoder	Decodificador de instrucciones
Instruction length	Longitud de una instrucción, tamaño de una instrucción
Instruction repertoire	Repertorio de instrucciones
Instruction set	Juego de instrucciones
Intelligent terminal	Terminal inteligente
Interactive	Interactivo
Interface	Interface
Interpreter	Intérprete
I/O	I/O
IPS	IPS
IRS	IRS
ISAM	ISAM

J

JCL	JCL
Joystick	Palanca de mando, palanca de control
Jump	Bifurcación
Justification	Justificación

K

K	K
Kernel	Núcleo
Key	Clave, tecla, llave
Keyboard	Teclado
Keyword	Palabra reservada
Kilobaud	Kilobaudio

Kilobit	Kilobit
Kilobyte	Kilobyte
Kit	Kit

L

Label	Etiqueta, rótulo
Laser printer	Impresora de láser
Library	Biblioteca
LIFO	LIFO
Light pen	Lápiz óptico
Line	Línea
LISP	LISP
Listing	Listado
Literal	Literal
Load	Cargar
Loader	Cargador, programa cargador
Logic programming	Programación lógica
Logic unit	Unidad lógica
Logical operation	Operación lógica
LOGO	LOGO
Loop	Bucle
Low-level language	Lenguaje de bajo nivel
LPS	LPS
LSI	LSI
LT	LT

M

Macroassembler	Macroensamblador
Macroinstruction	Macroinstrucción
Macrostatement	Macrosentencia
Machine	Máquina
Machine language	Lenguaje máquina
Magnetic card	Tarjeta magnética
Magnetic reader	Lector de caracteres magnéticos
Magnetic tape	Cinta magnética
Magnetic tape reader	Lector de cintas magnéticas
Magnetic tape unit	Unidad de cinta magnética
Mailing	Mailing
Main program	Programa principal
Maintenance	Mantenimiento
Mantissa	Mantisa
Mark	Marca
MARK 1	MARK 1
Mask	Enmascarar, máscara
Mass memory	Memoria de masa
Master card	Tarjeta perforada maestra
Matrix printer	Impresora de matriz de agujas
Mega	Mega
Megabyte	Megabyte, megaocteto
Megacycle	Megaciclo
Megahertz	Megahercios
Memory	Memoria
Memory location	Posición de memoria
Memory map	Mapa de memoria
Memory read	Lectura de memoria
Memory size	Tamaño de memoria
Menu	Menú
Microcomputer	Microcomputadora
Microelectronics	Microelectrónica
Microprocessor	Microprocesador
Microprogram	Microprograma
Microprogramming	Microprogramación
Microsecond	Microsegundo
Mini floppy disk	Minidiskette
Minicomputer	Minicomputadora
Minidisk	Minidisco
Modem	Módem, modulador, demodulador
MODULA	MODULA
Module	Módulo
Monitor	Monitor
MOS	MOS
MP/M	MP/M
Multiplexer	Multiplexor
Multiplexing	Multiplexado
Multipost	Multipuesto
Multiprocessing	Multiproceso
Multiprogramming	Multiprogramación
Multitasking	Multitarea
Multiuser	Multiusuario

N

NAND	NAND
Nano	Nano
Nanosecond	Nanosegundo
NE	NE
Network	Red
NMOS	NMOS
Node	Nodo
Noise	Ruido
Non volatile memory	Memoria permanente
NOR	NOR
Numeric keypad	Teclado numérico

O

Object code	Código objeto
Object language	Lenguaje objeto
Object program	Programa objeto
Octal	Octal
Off line	Fuera de línea
On line	En línea
Open	Abrir
Operand	Operando
Operating system	Sistema operativo
Operator	Operador
Optical fiber	Fibra óptica
Optical mark reader	Lector óptico de caracteres
Optimisation	Optimización
OR	OR
OR exclusive function	OR exclusivo
OR inclusive function	OR inclusivo
Output	Salida

P

Package	Paquete
Paper-tape punch	Perforadora de cinta de papel
Paper-tape reader	Lector de cinta de papel
Paper throw	Salto de papel
Parallel trasmission	Transmisión en paralelo
Parameter	Parámetro
Parity	Paridad
Parity check	Control de paridad
Partition	Partición
PASCAL	PASCAL
Password	Palabra de paso
PC	PC
Peripheral	Periférico
Personal computer	Computadora personal
Pico	Pico
Picosecond	Picosegundo
Picture element	Pixel
PILOT	PILOT
Pin	Patilla
Pixel	Punto de una imagen
PL/1	PL/1
PL/M	PL/M
Plotter	Plotter
Portability	Portabilidad, transportabilidad
Power	Potencia
Power on	Puesta en marcha
Power supply	Fuente de alimentación
Power up	Puesta en marcha
Printer	Impresora
Priority	Prioridad
Problem oriented language	Lenguaje orientado al problema
Prosedure	Procedimiento
Procedure oriented language	Lenguaje orientado al procedimiento
Process	Procesar
Processing	Procesamiento, tratamiento
Processor	Procesador

Program	Programa
Program generating	Generador de programas
Program step	Paso de programa
Programmable memory	Memoria programable
Programmable read only memory	Memoria programable de sólo lectura
Programmer	Programador
Programming language	Lenguaje de programación
PROLOG	PROLOG
PROM	PROM
Prompt	Orientación
Protocol	Protocolo
Pseudo-instruction	Pseudoinstrucción
Punched card	Tarjeta perforada
Punching	Perforación

Q

Queuing theory	Teoría de colas
QWERTY	QWERTY

R

RAM	RAM
Random access memory	Memoria de acceso directo
Rate	Velocidad
Read	Lectura
Read only memory	Memoria de sólo lectura
Read protect	Protección contra lectura
Reader	Lector
Ready	Preparado
Real time	Tiempo real
Real time operating system	Sistema operativo en tiempo real
Record	Registrar, registro lógico
Record length	Longitud de registro
Relocatable	Reubicable
Relocate	Reubicar
Remote terminal	Terminal remoto
Repeat	Iterar
Report generating	Generador de listados
Response time	Tiempo de respuesta
ROM	ROM
Routine	Rutina
RPG	RPG
RTL	RTL
RTOS	RTOS
Run time	Tiempo de ejecución
R/W	R/W

S

Sampling	Muestreo
Scope	Osciloscopio
Screen	Pantalla
Screen read	Lectura de pantalla
Screen size	Tamaño de pantalla
Sector	Sector
Seek time	Tiempo de búsqueda
Segment	Segmento
Semantics	Semántica
Sentence	Sentencia
Sequence number	Número de secuencia
Sequential	Secuencial
Sequential access memory	Memoria de acceso secuencial
Sequential program	Programa secuencial
Serial transmission	Transmisión serial
Silicon	Silicio
Simulation	Simulación
Single step	Paso a paso
Skip	Salto
Slot	Ranura
Soft key	Tecla de función programable
Software	Software, logical
Sort	Ordenación
Source code	Código fuente
Source language	Lenguaje fuente
Speed	Velocidad
Stand alone	Autónomo
Static memory	Memoria estática
Store	Almacenar
Stored program	Programa almacenado
Structured language	Lenguaje estructurado
Structured programming	Programación estructurada
Subprogram	Subprograma
Subroutina	Subrutina
Subscript	Subíndice
Supervisor	Supervisor
Support	Soporte
Symbol	Símbolo
Symbolic language	Lenguaje simbólico
Synchronous	Síncrono
Synchronous data transmission	Transmisión síncrona de datos
System	Sistema
System memory	Memoria del sistema
System programmer	Programador de sistemas
System software	Software del sistema

T

Tab	Tabulación, tabular
Table	Tabla de datos
Task	Tarea
Telecomunications	Telecomunicaciones
Teleprocessing	Teleproceso
Teletype	Teletipo
Terminal	Terminal
Text window	Ventana de texto
Thermal printer	Impresora térmica
TI	TI
Time sharing	Tiempo compartido
Time sharing system	Sistema de tiempo compartido
TOD	TOD
Track	Pista
Transfer	Transferencia
Translator	Traductor
Transmission	Transmisión
Truncate	Truncar
Truth table	Tabla de verdad

U

UCSD Pascal	UCSD Pascal
UNIVAC 1	UNIVAC 1
UNIX	UNIX
User	Usuario
User memory	Memoria de usuario
User program	Programa de usuario
Utility	Utilidad
Utility program	Programa de utilidad

V

Variable	Variable
Vector	Vector
Verification	Verificación
Video	Vídeo
Virtual memory	Memoria virtual
Volatile memory	Memoria volátil
Volumen	Volumen

W

Waiting queue	Cola de espera
Winchester	Winchester
Window	Ventana
Word	Palabra
Word processing	Tratamiento de textos
Work station	Puesto de trabajo
Write	Grabar
Write-protect	Protección contra grabación

X

XENIX	XENIX
XON-XOFF	XON-XOFF

Agradecimientos

Expresamos nuestro pleno agradecimiento a todas aquellas entidades públicas y privadas que nos han prestado su colaboración en esta obra, poniendo a nuestra disposición material gráfico propio o permitiendo la realización de reportajes fotográficos en sus instalaciones:

Apple-España, S.A.
Burroughs, S.A.
Caixa de Pensions «La Caixa»:
- Centre de Càlcul
- Microteca de l'Obra Social
- Museu de la Ciència de Barcelona
- Obra Social de La Caixa de Pensions

Centro de Cálculo de la Universidad Politécnica de Barcelona
Commodore
Hispano-Olivetti, S.A.
Logic Control
Museu Tèxtil i d'Indumentària de Barcelona
NCR España, S.A.
Nixdorf Computer
Siemens, S.A.
Texas Instruments

Referencias fotográficas y de ilustración

A.G.E.-Fotostock; Aisa; Algersuari, J.M.; Archivo Océano; Archivo Vendrell; Fabbri, Editori; Firo-Foto; Fortuny, A.; Fototeca/Tony-Stone; Godo-Foto; Océano (Martínez Tajadura, J.C.); The Image Bank; Zardoya.

ÍNDICE

LA COMPUTADORA: UNA MÁQUINA PARA ENSEÑAR Y APRENDER

PROGRAMAS EDUCATIVOS

LA INTELIGENCIA ARTIFICIAL Y LA QUINTA GENERACIÓN

INSTRUCCIONES MÁS COMUNES DEL LENGUAJE BASIC

plan general de la obra

VOLUMEN 1

VOLUMEN 2

Periféricos
Introducción. Tipos de periféricos
Unidades de impresión de datos.
Unidades de soporte de información. Plotters y digitalizadores
Módems y adaptadores de línea
Lectores ópticos de caracteres. Scanners. Periféricos de control de presencia

VOLUMEN 3

Software
Software de base. Software aplicativo
Sistema operativo
Conceptos básicos en la programación
Algoritmos y organigramas. Sistemas de numeración
Organización de los datos
Lenguajes de programación
Los lenguajes assembler. Los lenguajes de alto nivel
El lenguaje BASIC
El futuro tecnológico científico y las computadoras
La robótica. Las computadoras y el espacio
Redes locales

VOLUMEN 4

La computadora: una máquina para enseñar y aprender
La computadora y la educación. La alfabetización computacional
Alfabetización funcional
Imagen y texto
Programas educativos
Criterio para conocer los programas. Reforzamiento de estructuras
Simulación. Otras técnicas computacionales
Programas actuales y programas para el futuro
La inteligencia artificial y la quinta generación
Un futuro difícil de predecir
Prototipos hoy, realidad cotidiana mañana. ¿Puede pensar una máquina?
La inteligencia artificial
La quinta generación de computadoras
Instrucciones más comunes del lenguaje BASIC
Instrucciones de definición de matrices. Instrucciones de asignación
Instrucciones de entrada de datos. Instrucciones de salida de datos
Instrucciones de control. Funciones del BASIC
Instrucciones de manejo de ficheros en disco. Instrucciones gráficas
Algunos ejemplos de programas escritos en lenguaje BASIC
Glosario. Diccionario inglés-español

OCEANO